Software-Ergonomie

Christiane Rudlof

Handbuch Software-Ergonomie

Usability Engineering



Christiane Rudlof

Handbuch Software-Ergonomie

Usability Engineering



Inhaltsverzeichnis



1 Vorbemerkungen

1 Vorbemerkungen

Zur Zielgruppe dieser Broschüre gehören in erster Linie alle am Thema interessierten Personen, die mit Software-Entwicklungsprojekten in Berührung kommen, z. B. unmittelbar betroffene Benutzer, Entwickler, Fachkoordinatoren, Fachkräfte für Arbeitssicherheit, Interessensvertreter, Fachabteilungsvertreter und Qualitätsbeauftragte, die wissen möchten, wie man mit einfachen Mitteln die Nutzungsqualität einer Software verbessern kann. Anderen Personen bietet diese Broschüre einen Überblick über das Thema.

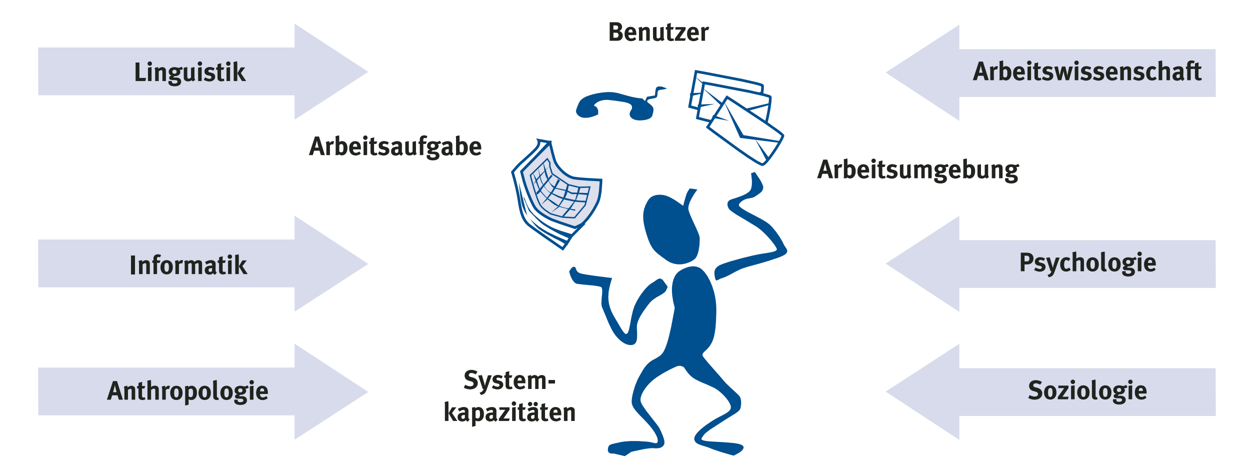
Die Broschüre umfasst zwei Themenkomplexe. Einerseits liefert sie detaillierte Gestaltungshinweise für ein benutzungsfreundliches Software-Produkt (Kapitel 3 und 4). Grundlage hierfür ist im wesentlichen die Normenreihe „Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten" DIN EN ISO 9241. Doch nicht für alle Entscheidungen, die während eines Entwicklungsprozesses getroffen werden, findet man die Lösung in dieser Normenreihe. Deshalb wird im Kapitel 2 und 8 der Prozess hin zu einem benutzungsfreundlichen Produkt aufgezeigt.
Ein Software-Produkt mit hoher Nutzungsqualität setzt einen benutzerzentrierten Entwicklungsprozess (Abschnitt 8.2) voraus, in dem bestimmte Methoden (Abschnitt 8.3) angewendet werden müssen.
Im Kapitel 2 wird auf verschiedene Typen von Software-Projekten eingegangen. Aufgezeigt wird, dass es unabhängig von der Art des Projektes möglich ist, eine der im Abschnitt 8.3 beschriebenen Methoden zur Verbesserung der Nutzungsqualität des Software-Produktes einzusetzen.
Je früher im Prozess diese Methoden angewandt werden, desto höher ist die Wahrscheinlichkeit, ein Software-Produkt mit hoher Nutzungsqualität zu erreichen.
Kapitel 5 befasst sich mit dem Thema Software-Ergonomie/Usability als Teil eines ganzheitlichen Software-Qualitätsverständnisses.

In gesonderten Abschnitten (8.5 und 8.6) finden sich Hinweise zu zwei Aspekten, die in fast allen Entwicklungs- oder Beschaffungsprojekten von Software eine Rolle spielen:

- die Kommunikation in den Projekten und
- die Bewältigung der Komplexität unter einem gegebenen Zeit- und Budgetrahmen.

Im Kapitel 9 schließlich finden sich Erläuterungen zur rechtlichen Situation.

75 % aller Software-Mängel werden während der Entwicklung verursacht. Davon bleiben 80 % bis zur Nutzung unerkannt.[1]



Software-Ergonomie ist die Optimierung des Zusammenspiels aller Komponenten, die die Arbeitssituation von Computernutzern bestimmen.

[1] SQS, Software Quality Systems, Köln

1 Vorbemerkungen

1.1 Von der Software-Ergonomie zum Usability Engineering

Ergonomie ist ein wissenschaftlicher Ansatz, damit wir aus diesem Leben die besten Früchte bei der geringsten Anstrengung mit der höchsten Befriedigung für das eigene und für das allgemeine Wohl ziehen.[2]

Der Begriff Ergonomie, also die Anpassung der Arbeitsbedingungen an die menschlichen Fähigkeiten und Fertigkeiten, stammt, wie das oben genannte Zitat, bereits aus dem vorletzten Jahrhundert.
Ging es damals um die Arbeitsplatzgestaltung an Maschinen, den körperlichen Einsatz und Umgebungseinflüsse, kam in den achtziger Jahren des letzten Jahrhunderts mit der Entwicklung der ersten PCs mit grafischer Oberfläche (GUI[3]) der Begriff Software-Ergonomie auf. Hierunter fasste man Erkenntnisse aus verschiedenen Fachgebieten zusammen, wie Psychologie, Arbeitswissenschaften, Informatik, Pädagogik, Anthropologie, Linguistik usw. In den Anfängen ging es vorwiegend darum, wie grafische Objekte bedient werden und um die Entwicklung der Schreibtisch-Metapher als Benutzungsoberfläche.
Im Detail gab es erste Ansätze, wie Daten auf dem Bildschirm sinnvoll nach verschiedenen Datenarten (Ein- und Ausgabedaten usw.) zu gliedern sind. Diese Überlegungen wurden bereits für zeichenorientierte Oberflächen angestellt.

Bald setzte sich die Erkenntnis durch, dass neben der Gestaltung der Oberfläche vor allem eine unangemessene Dialoggestaltung die Unzufriedenheit der Benutzer verursachte. Softwarebenutzung bedeutet in fast allen Fällen einen Eingriff in die Arbeitsgestaltung. Erkenntnisse darüber, wie Menschen arbeiten bzw. handeln, gab es von den Arbeitswissenschaftlern. An Arbeitsplätzen mit sicherheitskritischen Mensch-Maschine-Systemen, wie in Cockpits von Flugzeugen, Atomkraftwerken oder der Anlagensteuerung wurde relativ früh darauf geachtet, die Technik so zu gestalten, dass menschliche Fähigkeiten berücksichtigt sind.

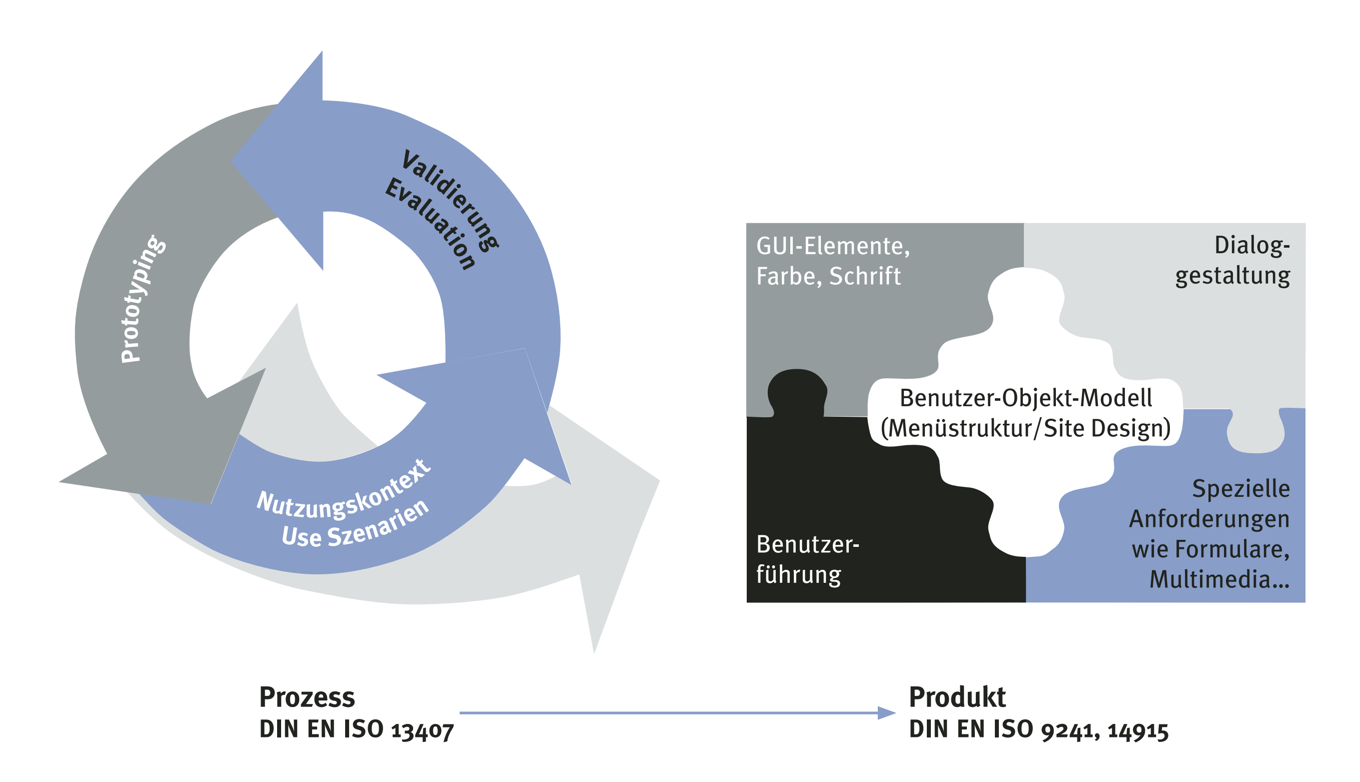
Dass auch im Bereich der (nicht sicherheitskritischen) Bürotätigkeit menschliches Verhalten zu berücksichtigen ist, fand seinen Niederschlag Ende der achtziger Jahre in der DIN 66234-10 (Ergonomische Dialoggestaltung).

Diese Norm war Vorläufer für die heute mehrteilige Normenreihe DIN EN ISO 9241, in der über 400 Empfehlungen für fast alle Bereiche der Software-Gestaltung zu finden sind. Es gibt jedoch keine einfachen Anweisungen, nach denen die Empfehlungen in der Norm anzuwenden sind. Vielmehr handelt es sich um kontext- und technologiefreie Regeln, die für jeden Anwendungsfall weiter konkretisiert werden müssen.

Softwareentwickler treffen jeden Tag zahlreiche Gestaltungsentscheidungen und erhoffen, einfache Antworten auf ihre Fragen zu bekommen. Erfahrungsgemäß ist es so, dass ein Usability Engineer selten einfache Antworten geben kann, sondern eher Fragen stellt: z. B. warum Daten so gruppiert sind, welche Daten in einer Auswahlliste zur Verfügung stehen sollen u. ä. Von den Antworten auf diese Fragen hängt dann ab, wie die entsprechende Normanweisung konkretisiert wird. Oft wissen die Entwickler auf diese fachlichen Fragen keine Antworten oder wenn man die „Fachseite" fragt, bekommt man unterschiedliche Aussagen aus unterschiedlichen Abteilungen. Dies ist der Punkt, an dem sich herausstellt, dass die Anforderungen nicht ausreichend genug mit Benutzern abgestimmt sind. Benutzer können zwar sagen, was sie benötigen, aber nicht, wie sie es technisch realisiert haben wollen. Dafür ist eine koordinierte Zusammenarbeit der Beteiligten erforderlich.

[2] W. Jastrzebowski, Grundriss der Ergonomie, 1857

[3] GUI = graphical user interface. Schnittstelle zwischen Benutzer und Anwendungsprogramm, die eine grafischbasierte Ausgabe (im Gegensatz zur zeichenbasierten Ausgabe) benutzt. Beispiel: Windows



Usability Engineering – eine prozessorientierte Methode

Ein gebrauchstaugliches Software-Produkt setzt einen benutzerzentrierten Entwicklungsprozess voraus. Dieser ist in der DIN EN ISO 13407, die seit 1999 gültig ist, grob beschrieben. Es geht also nicht mehr nur um die Fachdisziplin „Software-Ergonomie“, sondern es geht um das planmäßige Entwickeln von Benutzeranforderungen und das Integrieren in Software-Entwicklungsprojekte (Usability Engineering).

Festzuhalten bleibt, dass Usability Engineering ein Vorgehen ist, mit dem Ziel, Benutzererfahrungen in die Entwicklung von Produkten und Systemen einzubringen.

Es beinhaltet zwei fundamentale Elemente:

- multidisziplinäre Teamarbeit und
- die Anwendung einer Auswahl spezialisierter Methoden, um Benutzeranforderungen zu ermitteln und diese in Designvorschläge umzusetzen.[4]

Zusammenfassend kann man Usability Engineering definieren als:

Usability Engineering ist die Entwicklung von Nutzungsanforderungen, Prototypen und Softwareprodukten sowie deren Validierung und Verbesserung unter systematischer Anwendung von UE-Methoden im Rahmen des benutzerzentrierten Entwicklungsprozesses.

[4] Vredenburg, Isensee, Righi 2001, IBM User-Centered Design: An Integrated Approach.

1.2 Software-Ergonomie und Arbeitsschutz

Der Arbeitsschutz hat in Deutschland mit dem Arbeitschutzgesetz (ArbSchG) seit 1996 einen bedeutenden Wandel erfahren. Der Gesundheitsbegriff wurde neu definiert, als Zustand vollen psychischen, geistigen und sozialen Wohlbefindens.[5]
Im Rahmen des Arbeitsschutzes geht es hierbei u. a. um die Vermeidung psychisch belastender Arbeitsbedingungen. Arbeitsbedingungen werden als psychisch belastend verstanden, wenn

- sie den/die Arbeitende(n) bei der Aufgabenerledigung behindern und

- ihm dadurch überflüssigerweise zusätzlichen Handlungsaufwand abfordern.

Dies kann verursacht werden durch unübersichtliche, unstrukturierte und überladene Bildschirmmasken, die Verteilung von bei einem Arbeitsschritt benötigten Informationen auf unterschiedliche Masken/Anwendungen, durch fehlende, falsche oder unverständliche Fehlermeldungen, unklare Systemzustände bzw. Systemmeldungen, fehlende Möglichkeiten, Eingaben zu korrigieren, fehlende Möglichkeit des unmittelbaren Zugriffs auf häufig benötigte Masken, unangemessene automatische Cursorpositionierung oder umständlicher Wechsel zwischen unterschiedlichen Anwendungen. Weitere Beispiele kennen viele aus eigener Erfahrung.

Derartige arbeitsbezogene Belastungen können negativ auf das vegetative Nervensystem des Menschen wirken. Dieses besteht aus zwei sehr dünnen Nervensträngen rechts und links der Wirbelsäule und ist das vom Willen und Bewusstsein unabhängige Nervensystem, welches die Urlebensvorgänge vermittelt (z. B. das Atmen und Versorgen der Eingeweide und des Blutgefäßsystems). Bekannte Redewendungen, wie „das schlägt auf den Magen“, „das geht an die Nieren“ usw., kommen daher, dass man offenbar schon zu früheren Zeiten um diese Zusammenhänge wusste.

Arbeitsbezogene Belastungen zu vermeiden, ist also das Ziel gut gestalteter Software. Gleichzeitig erfolgt mit derart gestalteter Software auch eine Effizienzsteigerung, d. h. Produktivitätserhöhung.

Neben der Neudefinition des Gesundheitsbegriffs ist im Arbeitsschutzgesetz auch der Präventionsgedanke verankert. Ziel ist nicht mehr, ausschließlich wie vor 1996, die Vermeidung von Arbeitsunfällen, sondern zusätzlich die Gesunderhaltung der Arbeitnehmer. Daraus ergibt sich eben auch Handlungsbedarf bei Software-Anwendungen, und zwar bereits im Entwurfsstadium. Dieses ergibt sich aus Formulierungen in der mit dem Arbeitsschutzgesetz in Kraft getretenen Bildschirmarbeitsverordnung. Damit wurde der Anspruch von Arbeitnehmern auf „benutzungsfreundliche Software“ gesetzlich festgelegt.[6]
In der Bildschirmarbeitsverordnung wird gefordert, dass „bei der Entwicklung, der Auswahl, dem Erwerb und der Änderung von Software sowie bei der Gestaltung der Tätigkeit an Bildschirmgeräten der Arbeitgeber den Grundsätzen der Benutzungsfreundlichkeit Rechnung tragen muss.“

Die Chance, Software mit einer hohen Nutzungsqualität zu entwickeln, die damit sogar juristisch untermauert ist, gilt es zu nutzen.

[5] In Anlehnung an den Gesundheitsbegriff der Weltgesundheitsorganisation.
[6] Richenhagen, Prümper, Wagner: Handbuch der Bildschirmarbeit, 1997

1.3 Kein Sonderfall: Web[7]-Software und Multimedia-Anwendungen

Um das World Wide Web benutzen zu können, benötigt man eine so genannte Browser-Software[8] (browsing = herumstöbern, blättern). Die ersten Versionen dieser Browser waren nicht besonders benutzungsfreundlich. Erst die Entwicklung grafischer Browser-Oberflächen ermöglichte die Benutzung des World Wide Web nur mit Zeige- und Klickbewegungen, ohne sich umständliche Adressnamen merken zu müssen. Und dieses trug zu einer sehr schnellen weltweiten Verbreitung der Nutzung des World Wide Web bei. Der Unterschied zu herkömmlicher Software liegt in der Interaktivität, d. h. verschiedene Elemente (Grafiken, Text, Musik ...) interagieren miteinander. Dies geschieht hauptsächlich mit Unterstützung von Links.[9]

Bei einer Website kommt es auf drei Aspekte an: Inhalt, Form (Design) und Organisation (Struktur, Navigation). Nur wenn alle Aspekte miteinander abgestimmt entwickelt werden, kommt es zu einer qualitativ guten Website.

Entsprechend der Zielgruppe dieser Broschüre geht es hier unter der Bezeichnung Web-Software nicht um kommerzielle Online-Angebote. Obwohl für diese Usability geradezu zwingend ist, denn die „Konkurrenz ist nur einen Mausklick weit entfernt“. Das Thema wird hier im Abschnitt 4.8 lediglich aus dem Blickwinkel von Intranets betrachtet, die inzwischen wohl in jedem größeren Unternehmen vorhanden sind.

Wie bei konventionellen GUI-Applikationen ist ebenfalls zwischen der Oberflächen- und der Navigationsgestaltung zu unterscheiden. GUI-Applikationen verwenden oft Fenster- bzw. Dokumentenansichten zur Darstellung von Informationen.

Browser-Software sorgt jedoch sowohl für die Darstellung der Inhalte als auch für die Navigation zwischen den einzelnen Seiten.

Kritiker wie Jakob Nielsen[10] bemängeln dies und sagen einen Untergang der Browser voraus. Nach Ansicht von Nielsen gibt es keinen Grund, Daten abhängig von deren Speicherort Festplatte oder Internet[11] – unterschiedlich zu behandeln. Aber diese Sichtweise ist Zukunftsmusik.
Für die Oberflächengestaltung von Web-Software existieren inzwischen zahlreiche, öffentlich zugängliche Styleguides, an denen man sich orientieren kann (siehe Literatur).

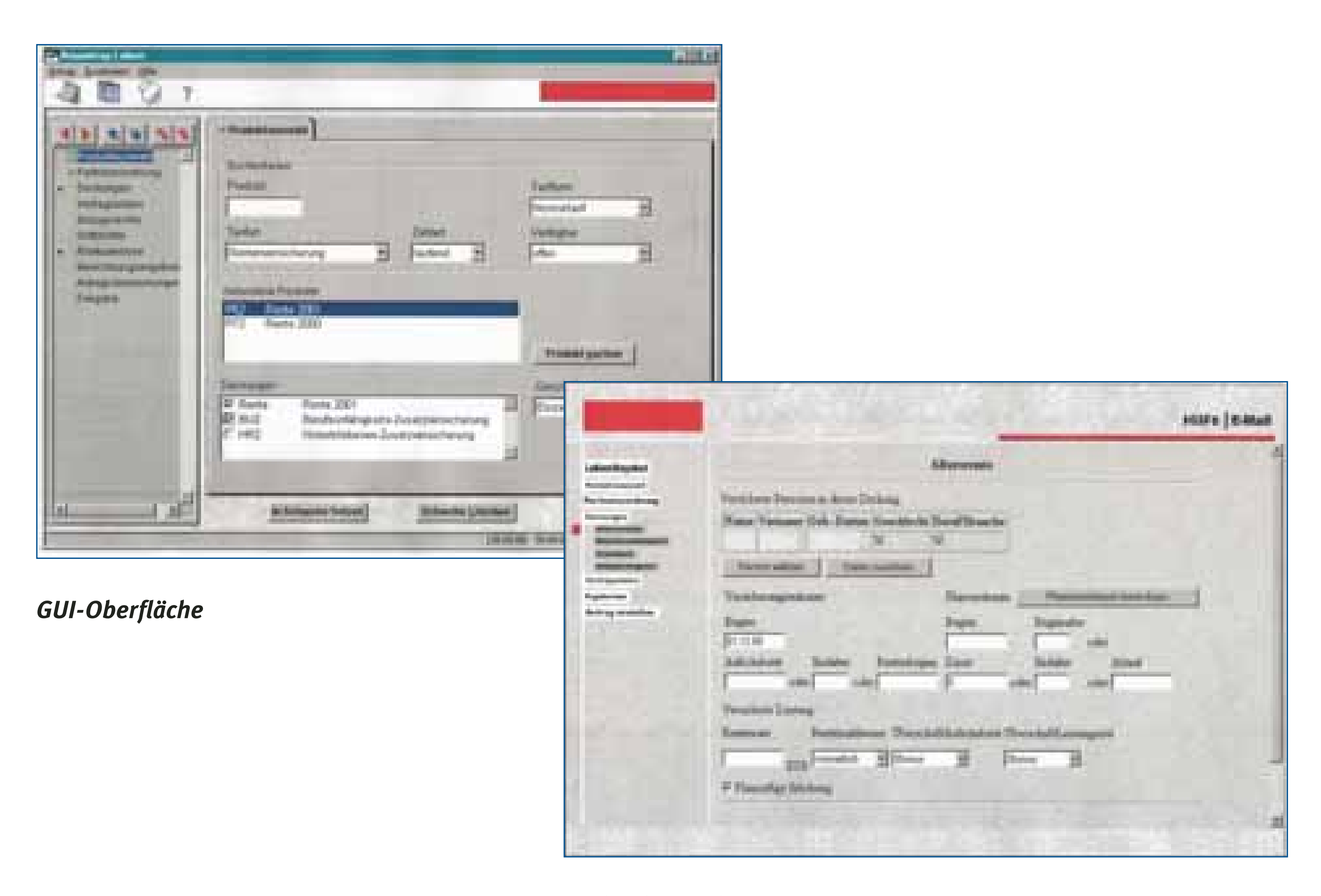
7 Kurzform für das World Wide Web, der Teilbereich des Internet, in dem spezielle World Wide Web-Server Dateien im HTML-Format zum Abruf durch Benutzer bereitstellen. Die grafische Oberfläche und einfache Bedienung führte zu einer weltweiten Nutzung.

8 Browser: Programme (mit grafischer Oberfläche) zum Aufrufen und Betrachten von Web-Seiten.

9 Link: Eine Verweisadresse im Internet auf eine andere Seite (Hyperlink = elektronischer Querverweis).

10 Nielsen, J.: Erfolg des Einfachen, 2000

11 Besteht aus einer Vielzahl von Computern, die das paketorientierte TCP/IP-Netzwerkprotokoll verwenden und direkt oder indirekt miteinander verbunden sind.



GUI-Oberfläche

Web-Oberfläche

Im Gegensatz zu konventioneller Software werden für Web-Anwendungen oft Styleguides erstellt. Dieses Dokument wird in der Regel von einer beteiligten Web-Agentur erstellt und enthält überwiegend gestalterische und technische Festlegungen, die sich z. B. auf das Corporate Design des Unternehmens beziehen. Die Nutzungsqualität für definierte Aufgaben im Intranet ist damit allerdings nicht garantiert. Diese Navigation in einem Hyperraum ist in jedem Fall eine kognitive Mehrbelastung, die es aus ergonomischer Sicht zu unterstützen gilt.

Diese Unterstützung kann nur erreicht werden, wenn Nutzergruppen und Szenarien definiert sind, für die diese Anwendung optimiert wird. Dies bedeutet, dass man sowohl kreative Aspekte als auch die Aspekte der Aufgabenunterstützung einbeziehen muss.

Bei der Entwicklung von Web-Software sind also einige Sichten mehr zu berücksichtigen, wie z. B. die Sicht der Marketing-Abteilung, wofür dann ggf. passende Methoden einzusetzen sind. Zur Erstellung eines Intranet mit hoher Nutzungsqualität sind jedoch Usability-Methoden (siehe Abschnitt 8.3) erforderlich.

1.4 Begriffsdefinitionen

Grundsätzlich ist zwischen zwei Begriffen zu unterscheiden: Benutzbarkeit und Gebrauchstauglichkeit.

In der Normung wird der etwas schwer verständliche Begriff für software-ergonomische Qualität verwendet: Gebrauchstauglichkeit (gemeint ist damit: für einen beabsichtigten Gebrauch tauglich).

Die DIN EN ISO 9241 Teil 11 definiert Gebrauchstauglichkeit als das Ausmaß, in dem ein Produkt

- durch bestimmte Benutzer
- in einem bestimmten Nutzungskontext genutzt werden kann,
- um bestimmte Ziele effektiv, effizient und mit Zufriedenheit zu erreichen.

Dies bedeutet:

„Bestimmte Benutzer" bedeutet, dass genauer spezifiziert werden muss (z. B. bzgl. Ausbildung, Anzahl, berufliche Erfahrung, IT-Kenntnisse, Alter).

Nutzungskontext bedeutet die Berücksichtigung der Benutzer, der Aufgaben, der organisatorischen Umgebung und der Einsatzumgebung (Hard- und Software, psychische und soziale Umgebung).

Aus den genannten Definitionen ergibt sich, dass es nicht die absolute ergonomische Qualität gibt. Qualität ist immer abhängig von den Anforderungen. In Orientierung an diese Anforderungen wird Software in bester Absicht von Software-Herstellern erstellt, d.h. die Software ist im Sinne der ISO/IEC 9126-1, ISO/IEC TR 9126-2, ISO/IEC TR 9126-3[12] benutzbar.
Die meisten benutzbaren Software-Produkte erweisen ihre Tauglichkeit aber erst im Nutzungskontext, dort stellt sich die Gebrauchstauglichkeit eines Produktes heraus, die Tauglichkeit für den Einsatzzweck am Einsatzort.

Gebrauchstauglichkeit ist also eine Nutzungsqualität und keine Produktqualität, die im Labor getestet werden kann.

Software-Produkte können nicht pauschal eine gute ergonomische Qualität haben, wenn man nicht weiß, welche Nutzer die Software für welche Aufgaben und in welchem Zusammenhang benutzen.

Mehr noch: es kann sogar sein, dass ein Software-Produkt zwar benutzbar, aber nicht gebrauchstauglich ist. Dies ist beispielsweise dann der Fall, wenn man mit einem Programm zwar Versicherungstarife berechnen kann, aber dies so umständlich zu handhaben ist, dass man „zu Fuß" schneller ist.
Der Unterschied zwischen Benutzbarkeit und Gebrauchstauglichkeit wird am Beispiel der abgebildeten Kaffeekanne deutlich.

Kriterium	Erläuterung	Definition laut DIN EN ISO 9241-11
„Effektiv erreichen"	Korrekte und vollständige Ausführung einer Aufgabe	Genauigkeit und Vollständigkeit, mit der Benutzer ein bestimmtes Ziel (Arbeitsergebnis) erreichen
„Effizient erreichen"	Erreichbarkeit und Aufwand zur Erreichung des Ziels	Im Verhältnis zur Genauigkeit und Vollständigkeit eingesetzter Aufwand, mit dem Benutzer ein bestimmtes Ziel (Arbeitsergebnis) erreichen
„Zufriedenheit"	Subjektiver Faktor	Beeinträchtigungsfreiheit der Nutzung und positive Einstellungen der Benutzer zur Produktnutzung

[12] ISO/IEC 9126 Software engineering – Product quality –



Benutzbarkeit:

Eine Menge von Merkmalen zur Erleichterung der Benutzung mit Rücksicht auf die individuelle Bewertung durch eine festgelegte oder vorausgesetzte Gruppe von Benutzern

Gebrauchstauglichkeit:

Die effektive, effiziente und zufriedenstellende Nutzung eines Produkts gemäß den Erfordernissen des Nutzungskontextes

A. Norman, The Design of Every Day Things, 1998

Sicherlich würde es gelingen, den Kaffee auszugießen, z. B. aus der Deckelöffnung statt über den Hals der Kanne. Es ist allerdings anzunehmen, dass dabei Kaffee verschüttet wird oder man sich sogar die Hand am heißen Kaffee verbrühen würde. Deshalb ist diese Kaffeekanne benutzbar (und sei es als kurioses Objekt für das Regal).

Wird jedoch als beabsichtigter Gebrauch definiert, dass man bequem über den Hals der Kanne den Kaffee in Tassen ausgießen möchte, ohne sich die Hände zu verbrühen und ohne Tropfspuren auf der Tischdecke zu hinterlassen, ist dies mit der Kanne nicht möglich.

Der beabsichtigte Gebrauch muss also definiert werden. Je genauer er beschrieben ist, desto höher ist die Wahrscheinlichkeit, dass das Produkt für einen beabsichtigten Gebrauch tauglich – im Sinne von „gebrauchstauglich“ ist.

Bei der Gestaltung von Alltagsgegenständen werden ergonomische Konzepte ebenfalls angewandt.



Zwei verschiedene Türgriffe nebeneinander an einem Auto. Die Funktion ist an der Gestaltung erkennbar.
links: senkrecht = schieben, rechts: waagerecht = ziehen A. Norman, The Design of Every Day Things, 1998

Die unterschiedliche Form der Türgriffe an dem Auto suggeriert dem Benutzer unmittelbar die Funktion dieser Griffe: in einem Fall wird die Tür aufgeklappt, im anderen wird die Tür aufgeschoben. Hier ist anhand der Form die Handhabung erkennbar. Vergleichbares findet sich bei den dreidimensionalen Drucktasten (Push Buttons) in Software-Produkten. Diese assoziieren durch ihre dreidimensionale Form, dass Benutzer sie „drücken" müssen, um eine Funktion auszulösen.

Design-Ergonomie

In Web-Agenturen besteht oft die Annahme, dass eine gut designte Anwendung automatisch benutzungsfreundlich ist. Design darf jedoch nicht mit Usability verwechselt werden. Beides sind Aspekte einer Web-Anwendung, jedoch mit unterschiedlichen Schwerpunkten. Design bezieht sich auf ästhetische Aspekte, Usability auf die Nutzungsqualität. Am besten macht diesen Unterschied folgendes Zitat deutlich:

> **„Ergonomie bedeutet, alles mit geschlossenen Augen zu finden. Design ist, wenn man trotzdem hinschaut."** [13]

[13] Aus einer Anzeige für den Skoda Fabia.

1 Vorbemerkungen

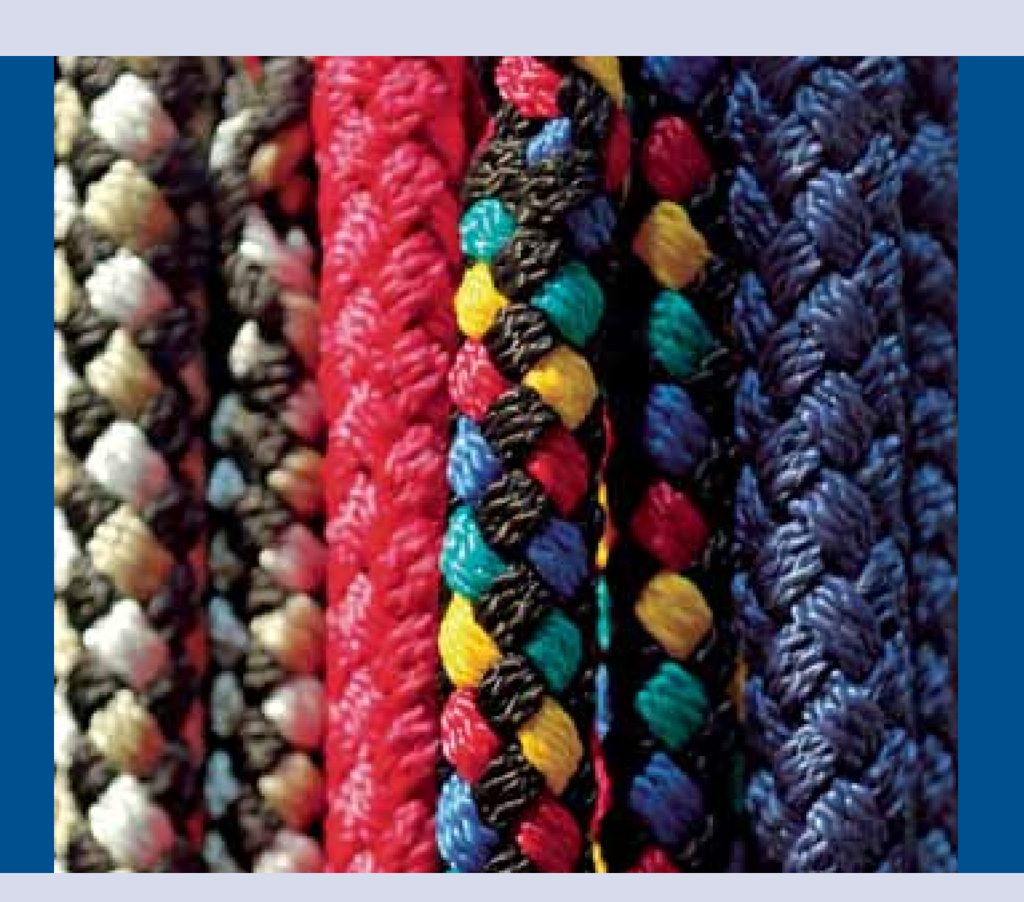


Design & Ergonomie im Kfz-Innenraum

Quelle: http://www.holzwurm-online.net/galerie/html

Jeder Autohersteller hat sein eigenes Designkonzept für den Innenraum, das ihn von der Konkurrenz unterscheidet. Es käme dabei aber wohl kein Hersteller auf die Idee, das Gas- und Bremspedal zu vertauschen. Hier gibt es eine Erwartungshaltung der Benutzer, über die man sich nicht hinwegsetzt.

Für Intranet-Anwendungen heißt dies, dass man beide Aspekte (Design und Usability) getrennt betrachten und dann miteinander abstimmen sollte. Ergebnis kann ein „Usability-Konzept“ und ein „Kreativ-Konzept“ sein (Abschnitt 8.3).



2

2 Von Softwaretypen und Projektarten

2.1 Einführung von Standardsoftware

2.2 Individualentwicklung

2.3 Neue Version (Release) einer bestehenden Software

2.4 Konzeption einer Intranet-Site

2 Von Softwaretypen und Projektarten

Es gibt nichts Gutes, außer man tut es.[1]

In der Bildschirmarbeitsverordnung[2] heißt es: „Bei Entwicklung, Auswahl, Erwerb und Änderung von Software sowie bei der Gestaltung der Tätigkeit an Bildschirmgeräten hat der Arbeitgeber den folgenden Grundsätzen – insbesondere im Hinblick auf die Benutzerfreundlichkeit Rechnung zu tragen."

In diesem Satz gibt es zwei Kernaussagen:

Die Formulierung „...sowie bei der Gestaltung der Tätigkeit" weist darauf hin, dass es einen Zusammenhang zwischen Softwaregestaltung und der Gestaltung der Arbeit (Tätigkeit) gibt. Und so wie es Kriterien für benutzungsfreundliche Software gibt, gibt es ebenfalls Kriterien für ergonomische Arbeitsgestaltung. Näheres dazu findet sich im Abschnitt 4.1.

Hier soll es um den ersten Teil des Satzes „Bei Entwicklung, Auswahl, Erwerb und Änderung von Software..." gehen. Diese Aufzählung bedeutet, dass unabhängig davon, um welche Art von Software-Projekt (Entwicklung, Kauf usw.) es sich handelt, die software-ergonomischen Mindestbedingungen zu erfüllen bzw. sogar einzufordern sind.

Im Folgenden soll anhand von Beispielen aufgezeigt werden, dass dieses tatsächlich möglich ist. Zwar wird im Abschnitt 8.2 ein komplexer, so genannter benutzerzentrierter Entwicklungsprozess beschrieben, dieser kann jedoch auch in Auszügen angewandt werden. Ohne Zweifel müsste man im Idealfall in Projekten bereits in der Analysephase beteiligt sein. Dies ist in der Praxis leider selten der Fall, dennoch sind auch zu einem späteren Zeitpunkt noch Gestaltungsmöglichkeiten durch den Einsatz entsprechender Methoden vorhanden.

2.1 Einführung von Standardsoftware

Standardsoftware[3] nennt man lauffähige Software-Produkte, die nicht für einen bestimmten Anwender programmiert sind und (betriebswirtschaftliche) Standardgeschäftsvorfälle (z. B. SAP R/3[4] Basissystem und Module, wie FI, CO) abbilden bzw. Standardarbeiten, wie Briefe schreiben (Microsoft „Word"), ermöglichen. Im Gegensatz zu einer Standardsoftware wie „Word" greift eine Software wie SAP R/3 bzw. die verschiedenen verfügbaren Module erheblich in die Unternehmensprozesse und Arbeitsabläufe ein. Es werden vor der Inbetriebnahme umfangreiche Anpassungsarbeiten notwendig. Diesen finanziell und zeitlich nicht unbedeutenden Teil einer SAP-Einführung nennt man Customizing[5]. Dabei geht es nicht nur um die Anpassung technischer Parameter, auch die Maskenfolgen werden hier bestimmt. Aus software-ergonomischer Sicht muss diese Projektphase durch eine benutzerzentrierte Anforderungsermittlung unterstützt werden. Ein Software-Hersteller wie SAP kann nur einen Nutzungskontext annehmen, für den die Benutzbarkeit gewährleistet wird.

Gebrauchstauglichkeit setzt jedoch voraus, dass ein konkreter Nutzungskontext und Use-Szenarien analysiert sind und die Anforderungen entsprechend konkretisiert und umgesetzt werden. In einer Untersuchung im Rahmen des „Ergusto-Projektes"[6] beurteilen Benutzer aus 15 Anwenderbetrieben die software-ergonomische Qualität des HR-Moduls sehr unterschiedlich. Das bedeutet, dass die Standardsoftware durch Customizing unterschiedlich gut den Arbeitsaufgaben angepasst werden kann, hier also in jedem Fall Einflussmöglichkeiten vorhanden sind.

Es sollten Methoden zum Einsatz kommen, wie z. B. die Ermittlung von Use-Szenarien (Abschnitt 8.3), um die Besonderheiten der Aufgabenstellung in einem Unternehmen oder einer speziellen Abteilung zu ermitteln. Da das Customizing meist durch interne Entwickler erfolgt, ist ein Benutzer-Entwickler-Workshop sinnvoll, ebenso wie ein Styleguide für die Gestaltung der zahlreichen Masken.

1 Erich Kästner
2 Bildschirmarbeitsverordnung (siehe Abschnitt 9.2.2)
3 auch ERP = Enterprise Ressource Planning (ERP) genannt
4 großes deutsches Software-Unternehmen für Standardsoftware
5 customizing (to costumize = anpassen) hier: Anpassung einer Software an die Bedürfnisse eines Kunden (Anwender)
6 Ergusto war ein vom Ministerium für Arbeit und Soziales, Qualifikation und Technologie in NRW (MASQT) gefördertes Projekt zum Ergonomischen Customizing von SAP R/3HR

2.2 Individualentwicklung

Bei einer Eigenentwicklung von Software in einem Unternehmen hat man optimale Chancen, von Beginn an benutzerzentrierte Methoden in die Projektplanung zu integrieren. Interessensvertreter z. B. sollten dieses auch offensiv der Geschäftsleitung anbieten. Der Vorteil der rechtzeitig koordinierten Einbeziehung von repräsentativen Benutzern und Usability-Experten ist, dass Streitereien vermieden werden und die Entwicklung beschleunigt wird. Gelingt es, nur eine „Prototyping Runde"[7] in die Planung einzubauen, wird es im Produktivbetrieb wesentlich weniger Nachbesserungen geben. Wichtig ist hier immer eine externe Moderation, da viele Diskussionen zu sehr von internen Sichten geprägt sind, die mit der eigentlichen Software-Entwicklung gar nichts zu tun haben.

Im Fall, dass intern die Anforderungen ermittelt und definiert werden, die Entwicklung aber an einen externen Software-Hersteller vergeben wird, gibt es ebenfalls Einflussmöglichkeiten. Diese sollten dann allerdings vertraglich fixiert werden. Z. B. sollte der Hersteller verpflichtet werden, zu einem bestimmten Zeitpunkt Prototypen zur Verfügung zu stellen und sich an der Lösungsfindung für Benutzeranforderungen aktiv zu beteiligen. Es muss geklärt werden, nach welchen Kriterien die vertragliche Abnahme des Software-Produktes durch den Kunden erfolgt. Oft verhindern ergonomische Mängel die Abnahme nicht, auch wenn die Mängel erheblich sind. Allerdings würde eine Abnahmeverweigerung voraussetzen, dass die ergonomische Qualität ausreichend konkret beschrieben ist. Qualität ist das, was gefordert wird. Es reicht nicht, vertraglich zu vereinbaren, dass die DIN EN ISO 9241-10 eingehalten werden soll, sondern die abstrakten Normformulierungen müssen in konkreter Form Vertragsinhalt sein.

2.3 Neue Version (Release) einer bestehenden Software

Bei der Planung einer neuen Version (Release) für eine bereits produktiv eingesetzte Software bietet es sich an, die bisherige Software auf Benutzungsfreundlichkeit zu überprüfen. Dies kann in Form von Benutzerbefragungen stattfinden. Diese können etwas über die subjektive Zufriedenheit aussagen oder Verbesserungsvorschläge ergeben. Eine andere Form ist ein Expertenreview[8], falls die Anwendung in der ersten Version ohne Usability Know-how erstellt wurde. Ein Experte kann Inkonsistenzen beim Einsatz von GUI-Elementen, Fehler im Navigationskonzept oder ähnliches feststellen. Der Software-Ergonomie Experte ist jedoch nicht Experte für die Arbeitsaufgabe. Deshalb müssen Benutzer immer mit einbezogen werden. Benutzer werden Aussagen darüber machen, warum sie welche Arbeitsschritte nicht so ausführen können, wie sie es bevorzugt täten. Beide Erkenntnisse zusammen geben zahlreiche Hinweise für die Weiterentwicklung der neuen Version.

2.4 Konzeption einer Intranet-Site

Bei der ersten Konzeption einer Website, an dieser Stelle ist damit ein Intranet für Beschäftigte eines Unternehmens gemeint, gibt es ähnlich viele Möglichkeiten, wie bei der Neukonzeption einer Individualentwicklung.

Unternehmen erhoffen sich mit der Einführung eines Intranets in erster Linie eine Reduktion interner Kosten. Ob dieses Ziel erreicht wird, hängt z. B. davon ab, ob ein Urlaubsantrag über das Intranet wirklich schneller abgewickelt wird als bisher. Kann das Intranet in solch elementaren Situationen nicht zu einem „Erfolgserlebnis" verhelfen, fehlt auch das Vertrauen in komplexere Vorgänge.

Deshalb erspart die frühzeitige Anwendung von Usability-Methoden Kosten und vermindert die Unzufriedenheit der Benutzer.

Die Praxis zeigt, dass man für ein Intranet dessen permanente Pflege mit einplanen muss. Mindestens muss die Aktualität der Daten gewährleistet sein. Ansonsten wird die Akzeptanz und das Interesse der Beschäftigen schnell nachlassen. Gelingt es nicht, Usability-Methoden von Anfang an in den Entwicklungsprozess zu integrieren, kann man mit Umfragen unter den Beschäftigten die größten Schwachpunkte herausfinden und Verbesserungen im Rahmen des nächsten Relaunches[9] einfordern. Näheres hierzu findet sich im Abschnitt 4.8.

---

7 siehe Abschnitt 8.3

8 siehe Abschnitt 8.3

9 Neuauflage einer Website mit wesentlichen Änderungen





3 Das Software-Produkt: Visuelle Gestaltung

3.1 Lesen am Bildschirm ist visuelle Belastung

Es gibt insgesamt über 400 Gestaltungsregeln für Software-Produkte. Diese sind je nach Kontext anzuwenden. Es ist zu empfehlen eine grobe Unterteilung in "look" und "feel", auf deutsch in Informationsdarstellung (visuelle Gestaltung) und Dialoggestaltung (Arbeitsabläufe) vorzunehmen. Im allgemeinen lassen sich Festlegungen zur visuellen Gestaltung einfach festlegen. Die angemessene Dialoggestaltung muss jedoch oft erst ermittelt werden. Benutzerbeschwerden beziehen sich oft auf mangelhafte Dialoggestaltung. Näheres dazu im Kapitel 4.

Bildschirmarbeit ist für die menschliche (visuelle) Wahrnehmung anstrengend. Dies resultiert daraus, dass man über lange Zeit Informationen mit den Augen auf eine bestimmte Entfernung (in der Regel 60 cm) fokussiert. Dies verursacht eine Muskelanstrengung im Auge. Nur bei ca. 5 Meter Abstand ist die Augenmuskulatur in einem völlig entspannten Zustand.

Die Frage, wie Menschen am Bildschirm lesen, hat der amerikanische Web-Forscher Jakob Nielsen[1] mit dem radikalen Hinweis beantwortet, dass sie gar nicht lesen. Die Wahrnehmung von am Bildschirm dargestellten Texten ist am besten mit dem Begriff „scannen“ (überfliegen) zu beschreiben.

Lesen Anfänger noch Buchstabe für Buchstabe, um ein Wort zu verstehen, kann man bei geübten Personen feststellen, dass in Blicksprüngen gelesen wird, indem die Gestalt eines Wortes erfasst und so schneller gelesen werden kann.
Bis zu sieben Buchstaben eines Wortes werden parallel verarbeitet. Nur wenn das erfasste Wort nicht in den verstandenen Zusammenhang passt, springt der Blick zurück und liest genauer.

So werden von geübten Benutzern die Feldbezeichner in Formularen nur noch aus dem Augenwinkel wahrgenommen. Deshalb ist es wichtiger, die Eingabefelder bündig auszurichten und nicht die Feldbezeichner selbst.

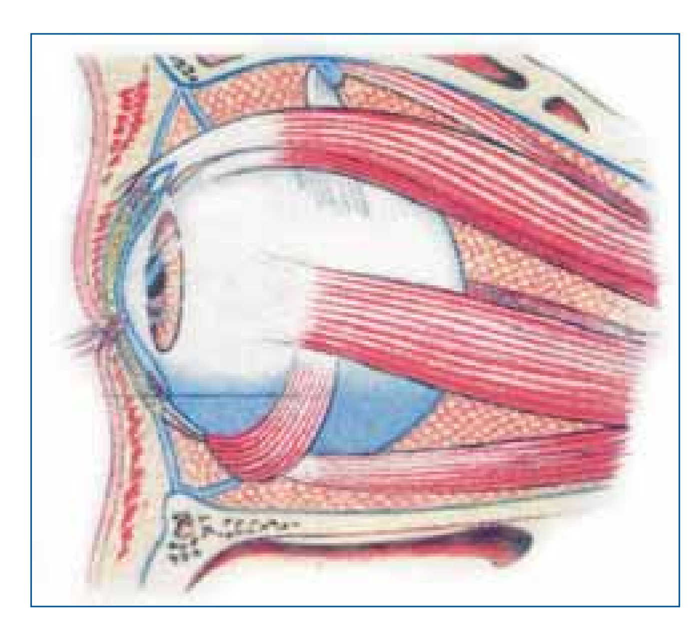
Ein weiterer Grund, warum Bildschirmarbeit für die Augen anstrengend ist, liegt darin begründet, dass die Darstellung auf Bildschirmen nicht so konturenscharf ist wie auf Papier, was aus den technischen Beschränkungen (Pixel pro Buchstabe) resultiert. Außerdem verursacht die Bildwiederholfrequenz der Kathodenstrahlbildschirme ein Flimmern der Darstellung.

Dies alles sind Gründe dafür, dass nachweisbar[2] am Bildschirm 30 % langsamer gelesen wird als auf Papier. Jeder Benutzer weiß außerdem, dass Schreibfehler oft erst in den Papierausdrucken entdeckt werden.

Insbesondere folgende Fähigkeiten des Auges spielen für die Informationsgestaltung am Bildschirm eine Rolle:

- Zum einen ist das Auge in der Lage, sich an verschiedene Entfernungen anzupassen (Akkommodation).
  Dies spielt z. B. für dreidimensionale Darstellungen oder auch für die Farbauswahl eine Rolle. Blaue Farbe wird als weiter entfernt empfunden als rot (chromatische Aberration).

- Zum anderen ist das Auge in der Lage, sich an verschiedene Lichtverhältnisse anzupassen (Adaptation). Bei Büroanwendungen sind in Anlehnung an Papier, überwiegend helle Hintergründe mit dunkler Schrift zu finden.



Schnitt durch das menschliche Auge

Weitere physiologische Fähigkeiten des Auges bzw. Sehsinns, die beansprucht werden:

- Räumliches und simultanes Sehen
- Koordination der Pupillenbewegungen in Abhängigkeit von der Stellung des Kopfes
- Brechkraft
- Farbsinn
- Gesamtkoordination der visuellen Wahrnehmung mit Körperbewegungen

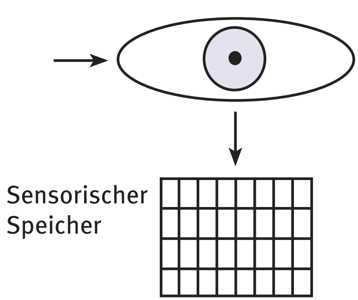
---

[1] http://www.nngroup.com

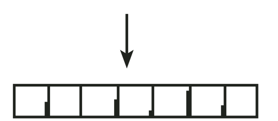
[2] Nielsen, J.: Designing Web Usability – Erfolg des Einfachen (2001), Verlag Markt + Technik



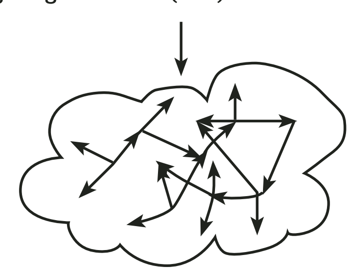
Reizaufnahme



Bei positiver Aufmerksamkeitserregung gelangen die Informationen ins Kurzzeitgedächtnis



Nach dem Assoziationsprinzip ins Langzeitgedächtnis (LGZ)



Drei Gedächtnisspeicher beim Menschen

Sensorischer Speicher	Das heißt für die Software-Gestaltung
Sensorisch: unverarbeitete physikalische Informationen Informationsmenge wird gefiltert und kategorisiert und mit Inhalt des Langzeitgedächtnisses (LGZ) verglichen, Inhalt geht nach max. 1,5 Sekunden verloren	Verwendung ergonomischer Farbkombinationen und anderer Kodierungen (z. B. Blinken)
Kurzzeitgedächtnis	
Ort bewusster kontrollierter Verarbeitung, begrenzte Kapazität (7 Einheiten), Komplexität von der Kodierung abhängig, speichert vor allem **symbolische** Daten, Speicherdauer inhaltsabhängig	Gruppierung von Informationen, Verwenden von Metaphern
Langzeitgedächtnis	
Unbegrenzte Speicherung (Kapazität und Dauer), enthält **episodische** und **semantische** Informationen, langsame Zugriffszeit	Konsistenz der Menühierarchie durch Strukturierung

Aber nicht nur das visuelle System des Menschen wird beansprucht. Auch kognitive[3] Faktoren spielen eine Rolle. Folgende Aspekte sind zu beachten:

Wahrnehmung: Wann und wie nimmt der Benutzer Informationen auf dem Bildschirm wahr?
Ein unterstrichenes „i" in der horizontalen Menüleiste für die Auswahl des Menüeintrags ist schwerer zu finden und zu lesen (weil schmaler) als die Unterstreichung eines „B".

Verständnis: Versteht der Benutzer, was auf dem Bildschirm geschrieben steht?
Eine Anzahl von Feldern mit der Überschrift „Adresse" lassen den Benutzer noch nicht erkennen, welches Feld für die Straße vorgesehen ist und welches für Ort und Postleitzahl.

Erwartungen: Was erwartet der Benutzer während der Bedienung des Systems?
Tastenkombinationen wie Strg+P (englisch: Print) für Drucken (statt Strg+D) sind nicht erwartungskonform und dadurch immer etwas konzentrationsintensiver. Der erhöhte Konzentrationsaufwand kann auch durch häufige Benutzung nicht reduziert werden.

Gewohnheiten: Was ist der Benutzer im Arbeitsalltag gewohnt?
Der Benutzer kann es gewohnt sein, sich zu jedem Kunden Notizen zu machen oder vorgangsbezogene Daten zur Bearbeitung an andere Kollegen weiterzugeben. Die Software sollte dies berücksichtigen.

Verhalten: Wie verhält sich der Benutzer mit dem Computer und welche Einstellungen hat er zur Software?
Er kann die Eingaben in Masken jederzeit abspeichern und auf Korrektheit überprüfen.

Gedächtnis: Was kann der Benutzer behalten?
Die Gedächtnisspanne beim Browsen durch Menüs reicht nicht weiter als ca. 4–5 Bildschirmseiten zurück. Dies hat man in so genannten „laufenden Gedächtnisaufgaben" herausgefunden. Dabei werden die Benutzer mit einer Sequenz von Bildschirmseiten und Menüs konfrontiert, die sie im Gedächtnis behalten sollen. Nach jeder neuen Seite werden Fragen zu den zuvor gesehenen Seiten gestellt. Informationen, die länger als 4–5 Seiten zurückliegen, werden kaum noch erinnert. (Im Gegensatz zum freien Browser im WWW konnten sich die Probanden allerdings nicht frei bewegen, z. B. um sich die Informationen auf den zuletzt gesehenen Seiten noch einmal anzuschauen.)

Erkenntnisse über diese Faktoren sind in die Empfehlung der DIN EN ISO 9241-12 eingeflossen. Kriterien für eine gute Informationsgestaltung sind in erster Linie eine klare Darstellung der Informationen (Schriftgröße etc.), eine gute Strukturierung der verschiedenen Datenarten und eine über alle Masken konsistente Darstellung. Viele Designentscheidungen sind jedoch Kompromisse. Hier muss man abwägen zwischen dem benutzerspezifischen Arbeitsablauf und der Konsistenz innerhalb der gesamten Anwendung.

3.2 Informationsdarstellung / Page Design (DIN EN ISO 9241-12)

Die folgenden Empfehlungen gelten sowohl für GUI-Masken als auch für das Layout von Intranet-Seiten. Der Mensch nimmt nicht alle Informationen auf einem Bildschirm gleichzeitig wahr, d. h. es werden nur kleine Informationseinheiten wahrgenommen. Durch Blickbewegungen muss daher eine Sequenz solcher Informationseinheiten entsprechend dem Arbeitsziel aufgenommen und verarbeitet werden.
Durch geeignete grafische Gruppierung und Formatierung der Informationen auf dem Bildschirm können logische Informationszusammenhänge vermittelt und belastende Suchbewegungen der Augen verringert werden. Als Anforderungen an die Informationsgestaltung werden in der DIN EN ISO 9241-12 die folgenden Kriterien benannt:

[3] kognitiv: die Erkenntnis betreffend

Kriterium	Beschreibung
Klarheit	Der Informationsgehalt wird schnell und korrekt vermittelt
Unterscheidbarkeit	Die angezeigten Informationen können gut unterschieden werden, z. B. Pflicht- und Kann-Felder
Kürze	Es wird nur so viel Information angezeigt, wie zur Aufgabenerfüllung nötig ist, z. B. keine unnötigen Wortverlängerungen bei Feldbezeichnern
Konsistenz	Gleichartige Daten werden durchgängig gleich dargestellt, z. B. Adressdaten in der Reihenfolge Name, Straße, Ort
Entdeckbarkeit/Auffindbarkeit	Die Aufmerksamkeit des Benutzers wird auf relevante Bereiche gelenkt, z. B. Platzierung wichtiger Daten im linken oberen Bereich
Lesbarkeit	Die Informationen sind leicht lesbar, z. B. durch ausreichende Schriftgröße und gut lesbare Schriftart
Verständlichkeit	Die Bedeutung der dargestellten Information ist leicht verständlich, unmissverständlich, erinnerbar, z. B. gut merkbare Icons

In der Norm finden sich zahlreiche konkrete Gestaltungsvorschläge:

Abschnitt der DIN EN ISO 9241-12	Es finden sich Gestaltungsempfehlungen zu:
Informationsorganisation	Fenster, Bereiche, Ein- und Ausgabebereiche, Gruppen, Listen, Tabellen, Felder
Grafische Objekte	Empfehlungen für die Objektgestaltung, Bewegungs- und Positionszeiger
Kodierung von Information	Alphanumerische Kodierung, Abkürzungen, Grafische Kodierung, Farbkodierung

Allein mit diesen Hinweisen würde man jedoch keine ergonomisch zufriedenstellende Maskengestaltung erreichen. Für die Organisation der Information auf den einzelnen Masken/Web-Seiten (Pages)[4] sind zuerst fachliche Überlegungen anzustellen, d. h. welche Daten gehören fachlich zusammen oder sind in welcher Reihenfolge darzubieten. Diese Überlegungen müssen sich auf die gesamte Anwendung beziehen. Es ist zuerst zu klären, welche Datenarten zu welchen Vorgängen gehören, ob es vorgangsübergreifende Daten u. ä. gibt. Es ist zu prüfen, bei welchen Daten es sich um Kerndaten und bei welchen Daten es sich um Zusatzinformationen handelt. Dies ist relevant für die Entscheidung, ob die Daten in normalen oder modalen Dialogfenstern dargestellt werden. Auch die Entscheidung über die Aufteilung innerhalb der Fenster, z. B. auf Registerkarten, innerhalb der Registerkarten in Gruppen, Tabellen usw. muss zuerst fachlich, d. h. ausgehend vom Arbeitsablauf, betrachtet werden.

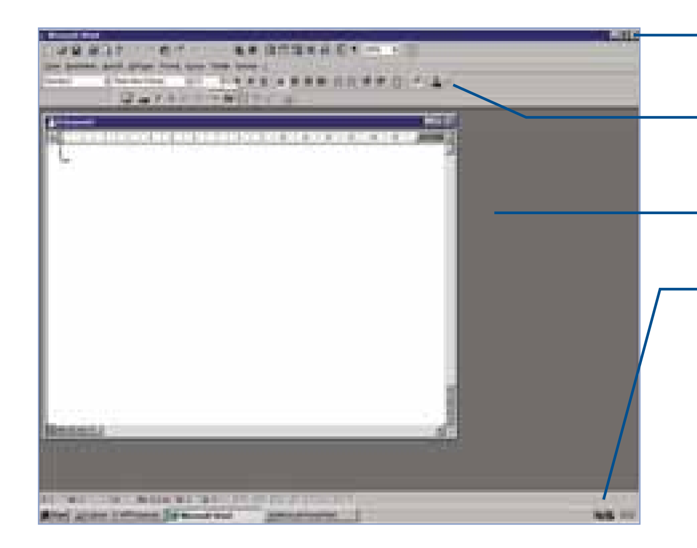
Ergänzend zu den fachlichen Überlegungen sind einige Faustregeln zu beachten:

- Auf den einzelnen Einheiten (Masken oder Reitern) sollten Gruppen von Datenblöcken gebildet werden. Datengruppen die auf verschiedenen Masken verwandt werden, sollten stets gleich angeordnet sein (z. B. Adressdaten, zuerst der Namen, dann die Straße, dann der Ort) und wenn möglich auch an gleicher Stelle positioniert sein (Ortskodierung). Auch der Aufbau von Tabellen oder die Zeilen/Spaltenanordnung sollte für gleiche Aufgaben gleich gestaltet sein.
  Für die Blockgruppen von Daten sollten in einem Styleguide Festlegungen getroffen werden, wie z. B.: max. 3–4 Blockgruppen pro Maske. Eine Blockgruppe besteht aus 2–3 Blöcken und diese wiederum aus maximal 10–12 Eingabefeldern.

- Die Unterteilung des gesamten Bildschirms in Arbeitsinformationen, Status- bzw. Systeminformationen und Steuerungsinformationen sollte über alle Masken konsistent sein. Die für Microsoft™-Produkte gängige Darstellung ist auf der Abbildung zu sehen (Seite 27 oben links). Zwar sind diese Einstellungen individuell änderbar, aber es gibt einen vorgeschlagenen Standard, der von den meisten Benutzern übernommen wird.

[4] Als Page bezeichnet man hier die einzelnen Seiten einer Web-Software. Alle Pages einer Anwendung zusammen nennt man Site.

™ Trademark, eingetragenes Warenzeichen



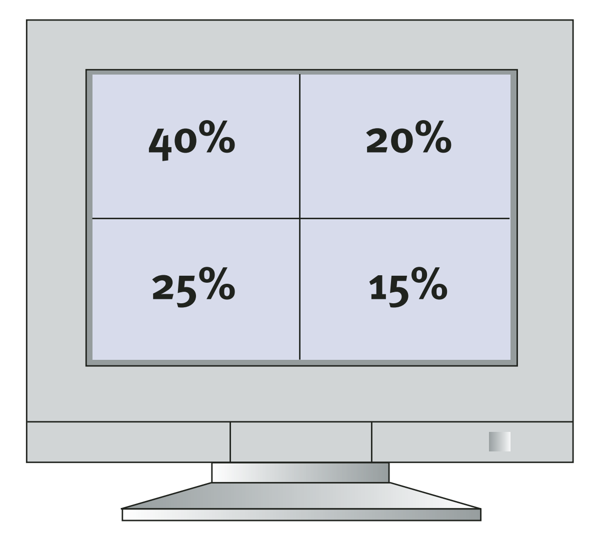
Einteilung des Bildschirms in Bereiche

Home Link zu anders-sprachiger Seite	Hilfe-Link Warenkorb
Login Wir über uns Produktkatalog Suchfunktion Sitemap	
FAQ Gästebuch/Forum	
Kontakt Impressum	

Einteilung einer Website in Bereiche

Quelle: eResult GmbH (http://www.eresult.de), August 2002

Bei der Platzierung einzelner Felder sollte berücksichtigt werden, dass die durchschnittliche Aufmerksamkeitsverteilung am Bildschirm sich wie in der Abbildung dargestellt verteilt:



Durchschnittliche Aufmerksamkeitsverteilung am Bildschirm[5]

Quelle: SAP

Dies bedeutet, dass man allgemeine Daten vor speziellen Daten, Pflichtfelder möglichst oben links anordnet und keine wichtigen Daten unten rechts positioniert.

- Die Bündigkeit als Gestaltungsprinzip benutzen, um unnötige Blicksprünge zu vermeiden.
- Die Bündigkeit von Eingabefeldern ist der Bündigkeit von Feldbezeichnern vorzuziehen, da der geübte Benutzer die Feldbezeichner nur noch aus dem Augenwinkel betrachtet.
- Zahlen sollten in Tabellen in der Regel linksbündig angeordnet sein.
- Spaltenüberschriften sollten analog zum Spalteninhalt angeordnet sein.
- So wenig Informationen wie möglich und soviel wie nötig auf einer Maske platzieren. Auch Freiflächen wirken als Strukturierung für die visuelle Wahrnehmung. Maximal 30-40 % des zur Verfügung stehenden Platzes nutzen.

[5] Gilt für Benutzer mit der Lese- und Schreibgewohnheit „von links nach rechts“ und von „oben nach unten“.

- Korrekte Verwendung von GUI-Elementen: Die Funktion bestimmt die Form, d.h. für eine Mehrfachauswahl werden Check Boxes, für eine Einfachauswahl werden Radio Buttons verwendet.

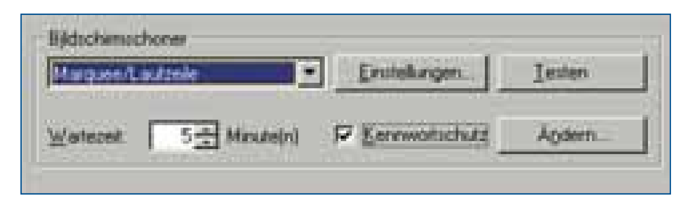
- Pixelgenau arbeiten, dies bedeutet, dass es ein Gestaltungsraster für die Masken geben muss, in dem die Abstände zu den Rändern, der Elemente untereinander usw. festgelegt ist.

ursprünglich

1. Textausrichtung an Unterkante
2-5. Standardisierte Abstände



verbessert



Quelle: http://www.sozialnetz-hessen.de/ergo-online

Anordnung von GUI-Elementen nach Tätigkeiten

- Kodierungen sind ein Mittel zum Hervorheben wichtiger Informationen. Mit Kodierungen[6] (Farbe, Fettschrift usw.) sollte man jedoch grundsätzlich sparsam umgehen. Kodierungsmöglichkeiten außer Farbe sind z. B. Helligkeit, Form, Ort, Umrandung, Gestalt, Blinken, Feldinvertierung.

- Zahlen sollten wie folgt gegliedert werden:
  Numerische Daten mit mehr als vier Ziffern ausreichend identifizierbar gliedern (Beispiele: 66 234, 14 363 434)
  Bevorzugt in Zweier-, Dreier- oder Vierergruppen gliedern.
  Bevorzugtes Gliederungszeichen ist das Leerzeichen.

- Vertraute Gruppierungen beibehalten, z. B.
  - Postleitzahlen (ausnahmsweise, weil inzwischen vertraut) als 5er-Gruppe:
    64285
  - Bankleitzahl im „BLZ-Format“:
    250 100 30

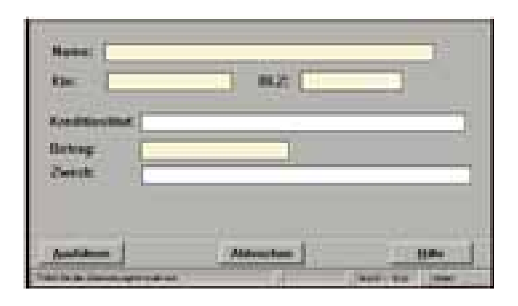
- Bei der Auflistung von Gegenständen, Eigenschaften usw. sollten vorzugsweise Ziffern als Ordnungsbegriff gewählt werden.

ursprünglich:	verbessert:
A. Größe	1. Größe
B. Gewicht	2. Gewicht
C. Farbe	3. Farbe

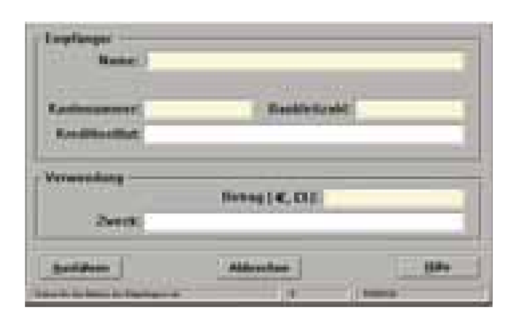
- Beim Prüfen aller Feldbezeichnungen, z. B. kommt es oft vor, dass in komplexen Anwendungen unterschiedliche Feldbezeichner für den gleichen Inhalt verwendet werden. Bei den Feldbezeichnungen sollte auf unnötige Wortverlängerungen verzichtet werden. Dies schafft mehr Platz auf den Masken. Auch die Positionierung der Feldbezeichner rechts- oder linksbündig oder oberhalb der Eingabefelder sollte einheitlich festgelegt werden. Dabei besteht das Ziel der Gestaltung darin, ein ruhiges Erscheinungsbild durch Einhalten von Fluchtlinien (gedachte Linien, an denen die Eingabefelder ausgerichtet werden) und identischen, pixelgenauen Abständen zu erreichen.

[6] Kodierung: Verschlüsselung von Daten mit Hilfe eines Kodes. Städtenamen werden z. B. durch Postleitzahlen verschlüsselt. Eine andere Möglichkeit der Kodierung sind Farben.

ursprüngliche Version



verbesserte Version



Weitere Optimierungen:
- *BLZ als gängige Abkürzung verwenden.*
- *Doppelpunkte weglassen.*
- *Trennlinie über Aktionsbuttons weglassen.*

Einhaltung von Fluchtlinien bei Eingabefeldern und Gruppierung von Daten

Um eine ergonomische und konsistente Maskengestaltung zu erreichen, empfiehlt sich die Anwendung eines Styleguide[7] als Qualitätssicherungsinstrument. In diesem werden alle getroffenen Entscheidungen dokumentiert.
Bei Software mit einer großen Anzahl von Masken ist es hilfreich, die Masken zu klassifizieren, z. B. Masken zur Dateneingabe, reine Infomasken usw. Für diese Maskenarten sollten dann Gestaltungsstandards festgelegt werden.

Prüfen kann man die Maskengestaltung , indem man die 4 „W“-Fragen stellt: Der Benutzer soll jederzeit sehen können:

- Wo bin ich?
- Wie kam ich hierhin?
- Was kann ich hier tun?
- Wohin und wie kann ich navigieren?

3.3 Die Gestaltgesetze

Die Gestaltpsychologie ist eine Wahrnehmungslehre. In den sog. Gestaltgesetzen wird formuliert, wie Menschen komplexe Szenarien oder Reize wahrnehmen, wie sie ihre Wahrnehmung gestalten. Die Umwandlung physikalischer Reize in anschauliche Gegenstände erfolgt nach bestimmten Gesetzmäßigkeiten, die vermutlich auf Erbstrukturen beruhen und kulturell übergreifend sind. Die Gestaltgesetze beziehen sich nicht auf Inhalte, sondern auf abstrakte Muster, Zusammenhänge und Eigenschaften.
Bekannte Konturen werden auch dann wahrgenommen, wenn sie nur in Bruchstücken vorhanden sind. Es gibt quasi einen Zwang zur Kontur.
In der menschlichen Wahrnehmung setzt sich immer eine Gliederung durch, die von den sog. „Ganzheitseigenschaften“ bestimmt ist.

Dies bedeutet:

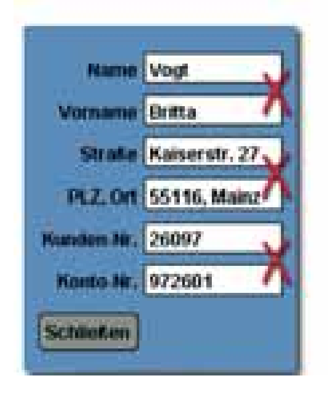
- Einfachheit
- Regelmäßigkeit
- Symmetrie
- Inneres Gleichgewicht

Durch die Gestaltung nach diesen Kriterien spart sich der Mensch das zeitaufwendige Punkt-für-Punkt-Abtasten einer Gestalt (bzw. der Informationen auf einer Bildschirmmaske), er erkennt die Information anhand ihrer äußeren Gestalt schnell wieder.

[7] siehe Abschnitt 8.3

Beispielhaft sind hier drei Gesetze und ihre Bedeutung für die Bildschirmgestaltung benannt:

Gesetz	Beschreibung	Bedeutung für die Informationsgestaltung	Abbildung
Gesetz der Nähe	In einer Menge gleichartiger Elemente schließen sich in unserer Wahrnehmung die räumlich nahe beieinander liegenden zusammen (auch wenn sie sich in Form, Größe, Farbe unterscheiden).	Logisch zusammengehörige Informationen werden auch örtlich zusammen gruppiert. Unterschiede in der Hierarchie o. ä. werden durch räumliche Trennung realisiert.	1 + 2
Gesetz der Gleichartigkeit	Bei Darbietung verschiedener Elemente werden gleiche oder gleichartige Elemente in einer Gruppe zusammengefasst wahrgenommen.	Zusammengehörige Informationen, z. B. Feldbezeichner, werden gleichartig dargestellt.	3
Gesetz der guten Gestalt	Der Mensch bevorzugt in seiner Wahrnehmung gute Gestalten (symmetrisch...)	Normalwerte werden als gute Gestalt dargestellt, Abweichungen verzerren diese Darstellung.	4









Gesetz der Nähe (Abbildung 1), Erläuterung: siehe Tabelle

Quelle: http://www.vbg.de/qualifizierung/lernprogramme.html



Vorher

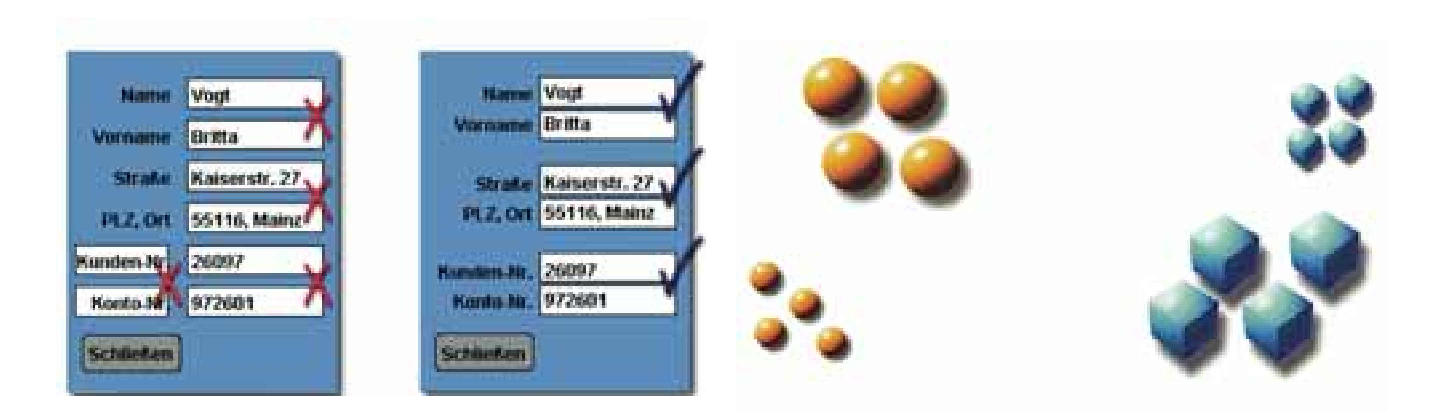
- *Die vertikalen Abstände der Kategorien sind hier verengt und auf gleiche Distanzen gebracht.*
- *Es wird weniger Fläche verbraucht, aber die Orientierung wird erschwert.*

Nachher

- *Die Links innerhalb der verschiedenen Gruppen bzw. Kategorien sind nach dem Gesetz der Nähe gruppiert.*
- *Die vertikalen Abstände zwischen den Kategorien (z. B. Finanzen, Sport) sind größer als die Abstände innerhalb einer Kategorie (Finanzen, Versicherung, Wirtschaft, Karriere).*

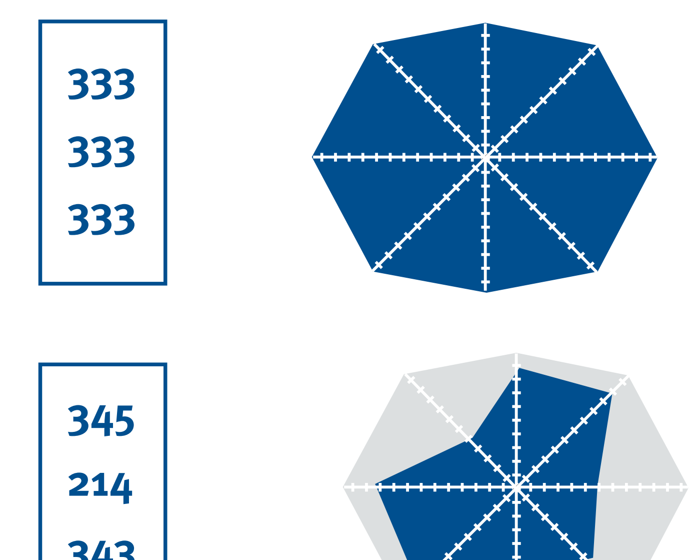


Anwendung der Gestaltgesetze bei WEB Content (Abbildung 2)



Gesetz der Gleichartigkeit (Abbildung 3), Erläuterung: siehe Tabelle

Quelle: http://www.vbg.de/qualifizierung/lernprogramme.html



Beispiel aus Echtzeitsystem:
Alternative Darstellung zu überwachender Werte als Parameter (links oben) oder als (gute) Gestalt (rechts oben).

Die Veränderung der grafischen Form (rechts unten) bei einer Veränderung der Parameter wird doppelt so schnell wahrgenommen wie die Zahlendarstellung (links unten).

Gesetz der guten Gestalt (Abbildung 4)

Das Berücksichtigen der Gestaltgesetze erspart unnötigen „Augenpfeffer", wie Trennstriche, Gruppenboxen usw. auf den Masken. Es wird quasi unsichtbar gestaltet.

Die Anwendung der Gestaltgesetze bei der Maskengestaltung bewirkt Folgendes:

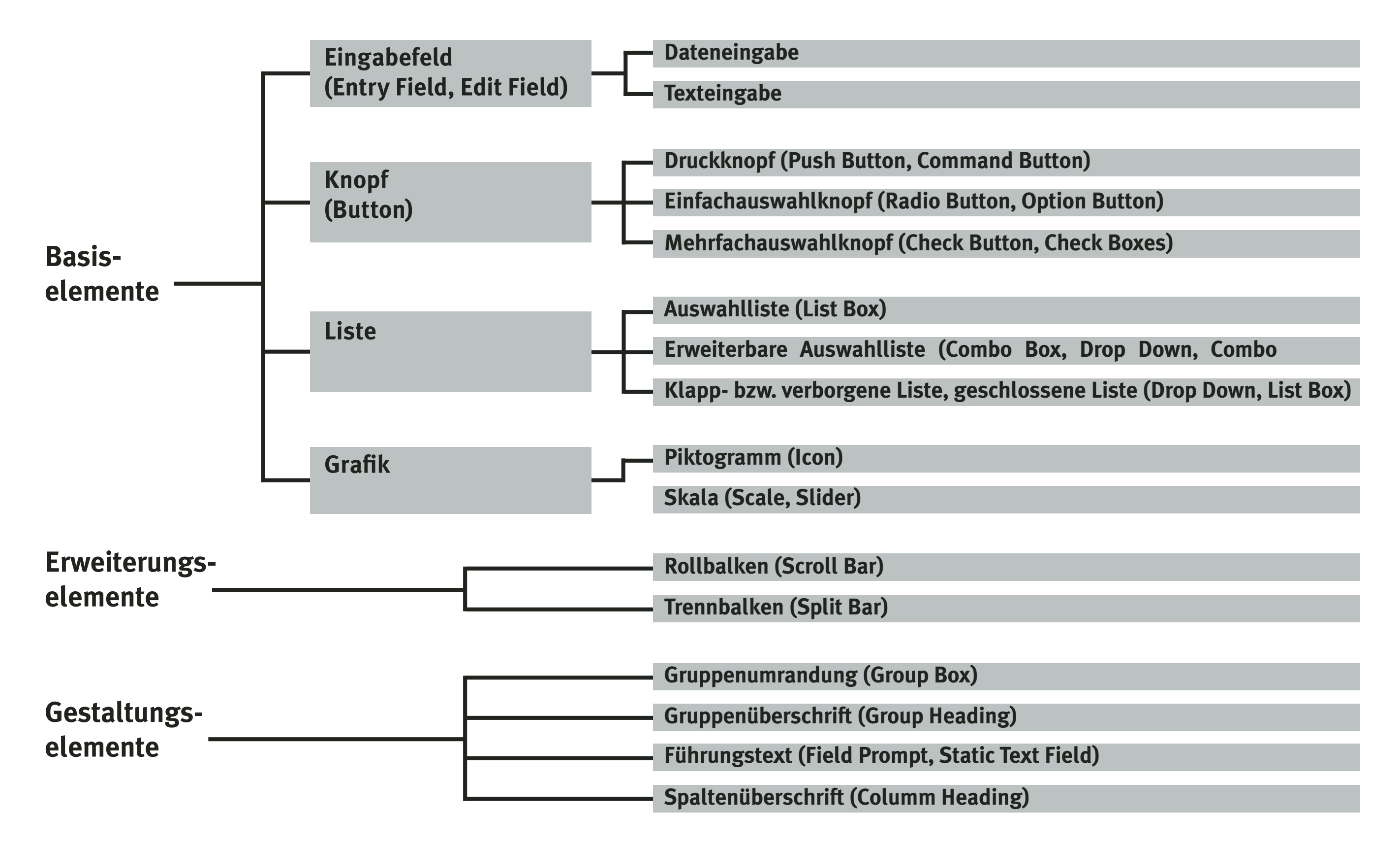
- erhöht die Wahrnehmbarkeit
- erleichtert das Suchen und Erkennen von Daten
- bietet ruhiges, ausgeglichenes Erscheinungsbild
- verringert Reaktions- und Entscheidungszeiten um bis zu 30 %

3.4 Verwendung von GUI-Elementen

Unter GUI-Element wird hier ein Interaktionsobjekt für grafische Benutzungsoberflächen verstanden, wie Eingabefelder oder Radio-Buttons. Eine umfangreiche Dokumentation dazu findet man im Windows-Styleguide[8] von der Firma Microsoft. Da viele Benutzer mit Microsoft-Produkten arbeiten, empfiehlt es sich, diesen Konventionen zu folgen, es sei denn, es gibt Gründe davon abzuweichen. Nicht zu empfehlen ist allerdings, einen grundsätzlich anderen Einsatz der Elemente zuzulassen, wie z. B. den Einsatz von Check Boxes als Navigationselemente zur Verzweigung auf andere Masken.
Weicht man von der Windows-konformen Verwendung eines GUI-Elementes ab, sollte man dieses konsistent in der gesamten Applikation durchhalten, denn uneinheitliches Verhalten von GUI-Elementen irritiert Benutzer erheblich.
Es kann vorkommen, dass die Verwendung der gängigen GUI-Elemente nicht ausreicht. Ein Indiz dafür kann sein, wenn sehr viel und lange über die Anwendung eines GUI-Elements diskutiert wird. Man sollte dann über die Use-Szenarien nachvollziehen, was der Benutzer an dieser Stelle tatsächlich tun muss. Dies kann dazu führen, dass es die beste Lösung ist, ein neues GUI-Element zu entwickeln, weil damit eine wesentliche Arbeitserleichterung erreicht wird. Es ist dann abzuwägen, was diese Entwicklung kostet (z. B. ein Entwickler benötigt einmalig 3 Tage, dafür sparen 100 Benutzer des neuen GUI-Elementes 0,5 Minuten pro Tag an Eingabeaufwand).

Auch wenn es allgemeine Konventionen über den Einsatz und das Verhalten von GUI-Elementen gibt, macht es in jedem Entwicklungsprojekt Sinn, einen Styleguide mit exakten Festlegungen als Mittel der Qualitätssicherung zu erstellen. Allerdings sollte der Styleguide neben der Verwendung der GUI-Elemente Aussagen zum Fensterkonzept, zur Navigation und zu den Dialoggrundsätzen enthalten. Der Styleguide kann je nach Projektumfang mehr oder weniger umfangreich sein. Für die Akzeptanz ist es wichtig, dieses Dokument mit den Entwicklern abzustimmen.
Für diese bedeutet es, dass sie sich an die Festlegungen halten müssen, es sei denn, sie können Abweichungen begründen.
Die folgenden Festlegungen können sich in dieser Form in einem Styleguide finden. Es handelt sich jedoch nur um Beispiele, von einer 1:1-Übernahme ist abzuraten. Die Aufzählung hat keinen Vollständigkeitsanspruch.
Auf der Abbildung ist eine Klassifizierung der gängigen Basiselemente zu sehen.

[8] Die Windows-Oberfläche, Leitfaden zur Softwaregestaltung, 1996



Elemente grafischer Oberflächen

3.4.1 Statischer Text



Verwendung
- Zur Bezeichnung von Überschriften (z. B. Titelzeile eines Dialogfensters oder Name eines Registers).
- Zur Bezeichnung von Dialogelementen, die der Eingabe oder Auswahl von Informationen bzw. Daten dienen (Radio Button, Check Box, Eingabefeld, Combo Box usw.).

Regeln
- Beginnt statischer Text mit einem Buchstaben, so ist ein Großbuchstabe zu wählen.
- Die Hintergrundfarbe von statischen Texten entspricht der Hintergrundfarbe des Fensters.
- Statischer Text, insbesondere wenn er der Beschriftung von Dialogelementen dient, wird nicht fett dargestellt.
- Der Zeilenabstand (die Höhe) ist so zu wählen, dass die Unterlängen einzelner Buchstaben (z. B. beim „g“) nicht abgeschnitten werden.
- Der statische Text wird linksbündig ohne Doppelpunkt vor das zugehörige Oberflächenelement positioniert.

3.4.2 Ein-/Ausgabefeld

Feldbezeichner

Verwendung
Das Eingabefeld wird für die Eingabe von Daten verwendet, deren Wertebereiche keinen oder nur geringen Einschränkungen unterworfen sind:

- Zahlen (z. B. Grad Celsius oder Geldbeträge)
- Freitext (z. B. Name oder Vorname)

Es gibt einzeilige oder mehrzeilige Eingabefelder.

Regeln
- Jedes Eingabefeld ist mit einer Beschriftung zu versehen.
- Für Eingabefelder, die der Eingabe eines Datums oder von Prozentzahlen dienen, sind Standardlängen zu verwenden.
- Wenn aufgrund von Platzproblemen nicht der vollständige Text angezeigt werden kann, so ist eine Scroll-Möglichkeit zur Verfügung zu stellen. Die Länge des Eingabefeldes muss sich an dem am häufigsten zu erwartenden Eingabewert orientieren.
- Ist die genaue Länge bekannt, z. B. bei Datumsfeldern, dann ist die Länge exakt zu wählen. Soweit sinnvoll und möglich sind Eingabefelder mit Vorschlagswerten zu belegen.
- Reine Ausgabefelder müssen gegen Eingaben geschützt werden.
- Inaktive Eingabefelder (= Ausgabefelder) sind mit derselben Hintergrundfarbe darzustellen wie der des Fensters. Die Hintergrundfarbe von „Kann-Felder“ ist weiß. Die Hintergrundfarbe von Pflichtfelder ist gelb.

3.4.3 Druckknopf (Push Button)

OK

Abbrechen

Verwendung
Push Buttons werden für folgende Aufgaben verwendet:

- Aufruf von modalen Dialogen
- Standardfunktionalitäten in Dialogen (also Sichern, Abbrechen etc.)
- Aufruf von anderen Dialogen

Regeln
- Für Fenster ohne Menü (z. B. eine Dialogbox) müssen alle Bearbeitungs- und Verzweigungsfunktionen mittels Push Button angeboten werden.
- Es ist immer ein Default[9] Button zu definieren.
- Push Buttons, die im aktuellen Kontext einer Bearbeitung nicht benutzt werden können, sind zu deaktivieren. Beispiel: „OK“ für Sichern. Dieser Button ist solange inaktiv, wie es nichts zum Sichern gibt.
- Werden in einem Fenster mit Scroll Bar Funktionen angeboten, die auf das gesamte Fenster wirken, so dürfen diese Push Buttons bei einem Scrollvorgang nicht mitscrollen.
- Push Buttons, die zu einer Datengruppe gehören, sind innerhalb der Gruppierung neben der Datengruppe zu positionieren.
- Wird über einen Push Button ein Dialogfenster aufgerufen, dann endet die Beschriftung mit 3 Punkten „...“.
- Push Buttons mit Textbeschriftungen müssen Mnemonics[10] haben.
- Die Push Buttons sind über die gesamte Anwendung hinweg einheitlich zu verwenden und anzuordnen.
- Die Bezeichnung ist zentriert auf dem Push Button angeordnet. Die Bezeichnung beginnt immer mit einem Großbuchstaben.
- Alle in einer Reihe angeordneten Push Buttons einer Maske sollten eine einheitliche Länge haben, d. h. die Länge richtet sich nach der längsten Bezeichnung.

[9] Default = Vorbelegung
[10] Mnemonic = leicht merkbare Abkürzung, um bestimmte Aktionen per Tastatur bedienen zu können

3.4.4 Einfachauswahlfeld (Radio Button)

Alternative1
Alternative2
Alternative3
Alternative4

Verwendung
Einfachauswahlfeld, Radio Button, Optionsschalter (alle drei Bezeichnungen werden verwendet) zur Auswahl von sich gegenseitig ausschließenden Alternativen, z. B. Auswahl weiblich, männlich. Nicht zulässig ist die Verwendung zur Abbildung von Bearbeitungs- und Verzweigungsoperationen.

Regeln
- Mindestens zwei Radio Buttons in einer Gruppe.
- Default Button bestimmen.
- Die Bezeichnung des Radio Buttons ist immer rechts daneben anzuordnen.
- Radio Buttons öffnen keine Dialogfenster oder lösen nicht direkt durch das An- oder Abwählen eine Funktion aus. Sie werden meist zur Darstellung von Zuständen verwendet.
- Radio Buttons können nebeneinander und/oder untereinander angeordnet werden.
- Bei vertikaler Anordnung sind die Schalter und Beschreibungen linksbündig untereinander anzuordnen.
- Es sollte zu jedem Radio Button ein Mnemonic angeboten werden.
- Bei mehr als sechs alternativen Optionen ist die Verwendung einer Auswahlliste zu prüfen.

3.4.5 Mehrfachauswahlkästchen (Check Box)

Einstellungen
Element 1
Element 2
Element 3
Element 4

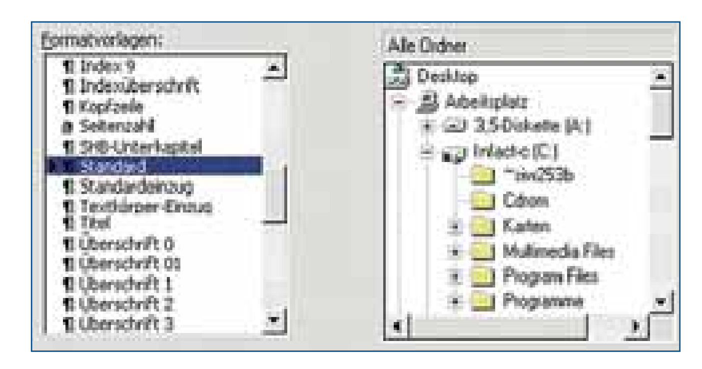
Verwendung
Mehrfachauswahlkästchen, Check Box, Kontrollkästchen (alle drei Bezeichnungen werden verwendet) zur Auswahl von sich nicht ausschließenden, logisch zusammengehörigen Alternativen. Check Boxen werden nur dort verwendet, wo ihr Inhalt keiner Dynamik unterliegt.

Regeln
- Die Bezeichnung ist immer rechts vom Quadrat zu positionieren.
- Bei Auswahl einer Check Box (aus einer Gruppe) bleibt der Zustand der anderen Check Boxen konstant.
- Sie öffnen keine Dialogfenster oder lösen nicht direkt durch das An- oder Abwählen eine Funktion aus. Werden mehrere Einstellungen angeboten, die inhaltlich zusammengehören, so sind die Check Boxes zu gruppieren.
- Werden mehr als acht Einstellungen angeboten, so ist alternativ eine Darstellung in Form einer List Box mit Mehrfachauswahl in Erwägung zu ziehen.
- Bei vertikaler Anordnung sind Kästchen und Bezeichnung linksbündig untereinander anzuordnen.
- Bei horizontaler Anordnung sind Kästchen und Bezeichnung auf gleicher Höhe anzuordnen.

3.4.6 Listen

Anmerkung: Je nachdem, welche Software zur Oberflächengestaltung verwendet wird, variieren die Bezeichnungen. Darüber hinaus gibt es Kombinationsformen. Wichtig ist es, den Verwendungszweck genau zu beschreiben und danach die passende Listenform zu bestimmen.

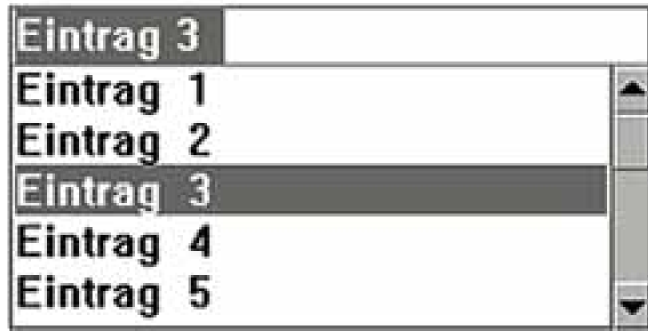
3.4.6.1 Auswahlliste (List Box)



Verwendung
Die List Box gehört zu den Dialogelementen, die zur Auswahl von gleichartigen Daten verwendet wird.
Die Auswahl kann verschiedenen Zwecken dienen:
- Zur Selektion von Datensätzen, die bearbeitet bzw. von einem Zustand in einen anderen überführt werden sollen.
- Mit Hilfe der List Box kann sowohl eine Einfach- als auch eine Mehrfachauswahl unterstützt werden.
- Die Menge der auszuwählenden Elemente ist in der Regel umfangreich und variabel (im Unterschied zu Radio Buttons).

Regeln
- Mindestens sechs bis acht Listeneinträge sollten direkt angezeigt werden. Bei Vergrößerung des Fensters sollte die Anzahl der direkt angezeigten Listeneinträge erhöht werden.
- Es gibt eine horizontale und/oder eine vertikale Scroll Bar, wenn die Auswahldaten nicht vollständig sichtbar und nicht direkt weitere Einträge in die List Box eingetragen werden.
- Die Einträge der Auswahlliste müssen inhaltlich miteinander verwandt sein.

3.4.6.2 Kombinationsliste (Combo Box)



Verwendung
Wenn eine variable Anzahl von vorgegebenen Auswahlalternativen angeboten werden soll, während andererseits ebenfalls die Eingabe eines individuellen Wertes (deshalb „Combo“ = Kombination) möglich sein soll.

Regeln
- Es ist eine horizontale und/oder eine vertikale Scroll Bar zu verwenden, wenn die Auswahldaten nicht vollständig sichtbar sind.
- Mindestens sechs Listeneinträge – soweit verfügbar – direkt anzeigen.
- Bei Vergrößerung des Fensters sollte die Anzahl der direkt angezeigten Listeneinträge erhöht werden.
- Nach Selektion eines Eintrags sollte die Combo Box automatisch wieder schließen.
- Einträge sollten inhaltlich miteinander verwandt sein.

3.4.6.3 Klappliste (Drop-Down List bzw. Drop-Down Combo Box)



Die Drop-Down List bzw. die Drop-Down Combo Box sind sowohl für die List Box als auch für die Combo Box die platzsparende Variante. Ist genug Platz auf der Maske, sollte dem Benutzer die Auswahlmöglichkeiten direkt angeboten werden, denn die Verwendung einer Klappliste erfordert für den Benutzer eine zusätzliche Aktivität (Liste zur Ansicht öffnen). Ist nicht ausreichend Platz, was eher der Normalfall ist, ist die Klappversion einzusetzen.

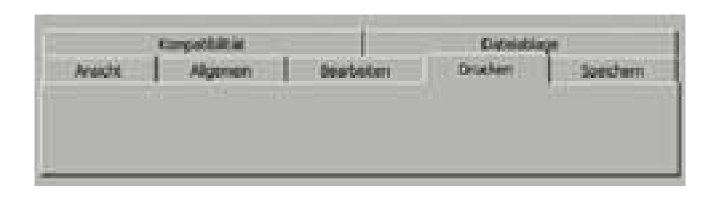
Verwendung

Zur Auswahl einer oder mehrerer Optionen, wenn innerhalb des Fensters nicht genügend Platz zur ständigen Darstellung aller Listeneinträge vorhanden ist oder eine Menge von Einträgen vorhanden ist, die fix bzw. selten verändert wird. Bei der Drop-Down Combo Box ist zusätzlich das aufklappbare Listenfeld mit einem Texteingabefeld kombiniert.

Regeln

- Es wird ein Initialeintrag innerhalb des Eingabefeldes angezeigt.
- Wenn kein spezifischer Initialwerte bestimmt werden kann, sollte der erste Listeneintrag angezeigt werden.
- Verwendung einer horizontalen und/oder vertikalen Scroll Bar, wenn die Auswahldaten nicht vollständig angezeigt werden können.
- Nach Selektion eines Eintrags automatisch wieder schließen.
- Im Rahmen des Aufklappmechanismus nicht mehr als sechs Listeneinträge gleichzeitig anzeigen.
- Die Einträge der Klappliste müssen inhaltlich miteinander verwandt sein.

3.4.7 Register (Tab Control)



Verwendung

Register werden dann verwendet, wenn die Dialogelemente einer Maske nicht mehr auf eine Bildschirmseite passen und eine entsprechende Gruppierung der Dialogelemente bzw. der Daten möglich ist. Entsprechende Arbeitsschritte müssen so gruppiert werden können, dass die Bearbeitung unabhängig voneinander erfolgen kann. D. h., um eine Bearbeitung abschließen zu können, darf kein Registerwechsel notwendig sein.

Regeln

- Die Breite der einzelnen Register sollte nach Möglichkeit ausgeglichen sein. Es ist darauf zu achten, dass die Breite der Register nicht stark variiert.
- Der Abstand zwischen der Bezeichnung eines Registers und dem darunter positionierten GUI-Element ist so zu wählen, dass eine optische Trennung zwischen beiden existiert.
- Als Titel von Registern sind grundsätzlich Texte zu wählen, d. h., Symbole sind unzulässig.
- Es ist eine sprechende Beschreibung zu wählen, d. h. Abkürzungen beispielsweise sind im Regelfall unzulässig.
- Die Register werden nach ihrer Wichtigkeit gruppiert. Das Register, das am meisten verwendet wird, ist links zu gruppieren, das Register, das am seltensten benötigt wird rechts.
- Bei Aufruf des Dialogfensters wird standardmäßig das Register angeboten, das am häufigsten benötigt wird, also im Normalfall das erste Register.
- Register müssen mit der Maus aktivierbar sein.

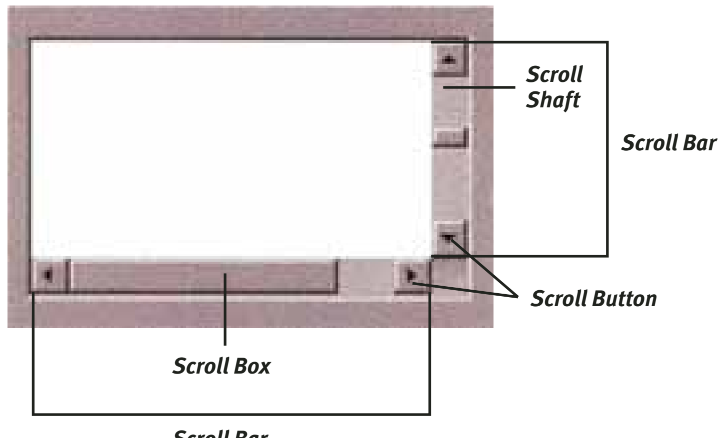
3.4.8 Tabellen



Verwendung
Tabellen dienen der Darstellung bzw. Auswahl von Datensätzen (z. B. aus einer Datenbank) in Form der zeilen- und spaltenweisen Anordnung. Im Normalfall dienen Tabellen der reinen Anzeige von Daten und nicht für deren Bearbeitung.

Regeln
- Es ist eine horizontale und/oder eine vertikale Scroll Bar anzubieten, wenn die Auswahldaten nicht vollständig sichtbar sind.
- Die Einträge der Tabelle müssen inhaltlich miteinander verwandt sein.
- Ein vergrößerter Zeilenabstand nach drei oder vier Zeilen erleichtert das Erkennen einer Zeile über mehrere Spalten (Gesetz der Nähe) besser als ein farbliches Absetzen.
- Beim Scrollen der Tabelle müssen alle Spalten synchron mitscrollen.
- Wird der Spaltenkopf einer Spalte angeklickt, wird die gesamte Liste nach dem Inhalt der selektierten Spalte sortiert, entweder in aufsteigender oder in absteigender Folge in Abhängigkeit der aktuellen Sortierung. Es wird zuerst immer aufsteigend nach der ausgewählten Spalte sortiert.
- Die Spaltenüberschriften sind sprechend zu wählen, d. h., die Übernahme der Spaltennamen von Datenbanktabellen ist im Normalfall nicht zu empfehlen.
- Für die Spaltenbreite ist – analog zu den Eingabefeldern – im Idealfall die Maximallänge zu wählen. Falls dies aus Platzgründen nicht möglich ist, so ist eine Breite zu wählen, die genügend Raum zur Verfügung stellt, um den Normalfall abzudecken.
- Falls nicht alle Spalten gleichzeitig darstellbar sind, so ist eine Priorisierung durchzuführen. Die Spalten werden gemäß ihrer Priorität angeordnet, die wichtigste(n) stehen links.

3.4.9 Bildlaufleiste (Scroll Bar)



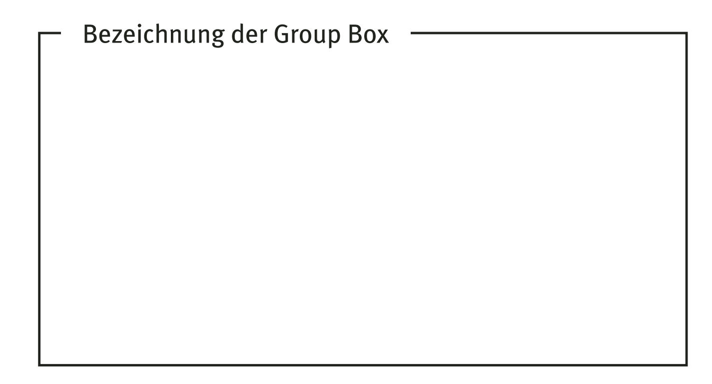
Verwendung
Sobald nicht alle Informationen eines Dialogelementes vollständig angezeigt werden können, ist eine horizontale bzw. eine vertikale Scroll Bar[11] anzubieten.

Regeln
- Eine Scroll Bar muss anzeigen, in welcher „Richtung“ weitere Informationen vorliegen. D. h. stehen weitere Informationen in der Breite zur Verfügung, so ist eine horizontale Scroll Bar anzubieten; stehen weitere Informationen in der Länge zur Verfügung, so ist eine vertikale Scroll Bar anzubieten.
- Sobald in eine „Richtung“ keine weiteren Informationen mehr vorliegen, ist der entsprechende Richtungs-Scroll Button inaktiv darzustellen.
- Die Größe der horizontalen Scroll Bar ist so auszurichten, dass Rückschlüsse auf das Verhältnis zwischen derzeit sichtbaren Informationen und gesamt verfügbaren Informationen gezogen werden können, d. h., wird derzeit die Hälfte aller verfügbaren Informationen angezeigt, so belegt die Scroll Box[12] die Hälfte des Bereichs zwischen den beiden Scroll Buttons.
- Die Position der Scroll Box hat die Position der derzeit sichtbaren Informationen im Verhältnis zu den unsichtbaren Informationen wiederzuspiegeln. D. h., wenn derzeit die ersten Informationen in dem Fenster angezeigt werden, steht die Scroll Box ganz oben oder ganz rechts.

[11] Scrollen = Verschieben der Ansicht eines Objektes, so dass andere Teile des Objekts sichtbar werden

[12] Scroll Box = Bestandteil der Bildlaufleiste, der die relative Position des sichtbaren Bildausschnitts zum gesamten Dokument darstellt

3.4.10 Gruppenumrandung (Group Box)



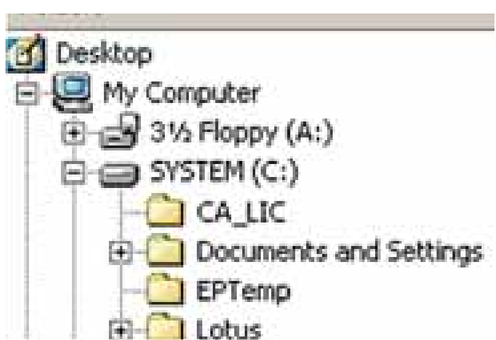
Verwendung

Die Gruppenumrandung wird verwendet, um Dialogelemente inhaltlich zusammenzufassen und von ihrer Umgebung abzuheben oder gegenüber anderen Dialogelementen abzugrenzen.

Regeln

- Eine Gruppe kann durch eine Umrandung gekennzeichnet werden.
- Group Boxen sollten sparsam verwendet werden und nur dann, wenn räumlicher Abstand als Gliederungsmittel nicht möglich ist.
- Jede Group Box muss eine Bezeichnung haben, die linksbündig auf der oberen Umrandungslinie ausgerichtet wird. Alternativ können Gruppen auch lediglich durch eine Gruppenbezeichnung (Group Heading) gekennzeichnet werden, die linksbündig ausgerichtet über der Gruppe dargestellt wird.

3.4.11 Baumdarstellung (Tree)



Baumdarstellungen bieten durch die Baumstruktur das beste Potential für eine dynamische Darstellungsform, mittels derer Informationen dargestellt werden können, für die eine hierarchische Abhängigkeit besteht. Der Benutzer kann die Feinheit der Darstellung für einen (Teil-) Baum seinen aktuellen Bedürfnissen anpassen, indem er ihn aus- bzw. einklappt. Ein Baum besteht aus Knoten und „Blättern“. Die Symbole dafür sind z. B. Ordner und Dateidarstellungen. Die Symbole werden durch Texte benannt.
Die Verwendung von Baumdarstellungen zur Anzeige oder für die Bearbeitung von Daten ist individuell für jeden Einzelfall abzuwägen und abzustimmen.

3.4.12 Kontextmenü



Verwendung
Kontextmenüs wirken auf Anwendungsobjekte. Sie dienen dem effizienten Zugriff auf die Operationen eines ausgewählten Objekts.
Die Menüpunkte eines Kontextmenüs sollten sich i. d. R. im Applikationsmenü wiederfinden. Ausnahmen bestätigen die Regel. Ein Beispiel hierfür ist der übliche Menüpunkt in Microsoft-Kontextmenüs mit der Bezeichnung „Eigenschaften".
Sie werden aktiviert durch Einfachklick mit der rechten Maustaste auf ein Objekt im Arbeitsbereich des Fensters oder für den Arbeitsbereich selbst, falls für diesen ein Kontextmenü spezifiziert wurde.
Der Einsatz von Kontextmenüs reduziert die Bewegung mit der Maus, die zur Funktionsauslösung ständig zur Menüleiste bewegt werden muss. Über das Kontextmenü ist eine Funktionsauslösung an Ort und Stelle möglich.

Regeln
- Der Einsatz von Kontextmenüs ist sinnvoll, wenn einem Objekt gezielt Funktionen zugeordnet werden sollen.
- Die Kaskadierung von Kontextmenüs sollte nur im Ausnahmefall und nur über eine Hierarchieebene erfolgen.
- Die Menüeinträge im Kontextmenü beziehen sich stets auf das angeklickte Objekt oder den Arbeitsbereich, in dem das Kontextmenü aufgerufen wird.
- Im Kontextmenü dürfen nur die wichtigsten objekt- und arbeitsbereichsbezogenen Funktionen als Menüeintrag enthalten sein.
- Die Breite des Kontextmenüs entspricht dem des breitesten Menüeintrags.
- Das Menü wird automatisch geschlossen, wenn ein Menüeintrag ausgewählt wird oder wenn der Benutzer eine andere Funktion mit der Maus oder der Tastatur wählt.
- Menüeinträge, die in der aktuellen Dialogsituation nicht wählbar sind, werden in vermindertem Kontrastniveau dargestellt (deaktiviert).
- In der Regel finden sich im Hauptmenü alle Funktionen, im Kontextmenü die im jeweiligen Kontext wichtigsten.

3.5 Verwendung von Schrift und Farbe

Bei der Auswahl von Schrift kommt es auf mehrere Faktoren an: Die Zeichen müssen gut erkennbar sein, Verwechslungen einzelner Buchstaben oder Zahlen sollten ausgeschlossen sein.

Als grundsätzliche Empfehlung gilt:

- Die Schrift sollte größer sein als auf Papier.
- Empfehlenswert sind mindestens 10 Punkt-Schriften, besser 14 Punkt-Schriften. [13]
- Vorschrift aus der Normbestimmung sind 3-4 mm (7x9 Punkt Matrix)[14] für Großbuchstaben ohne Oberlängen.
- Auf Papier ist eine Schrift mit Serifen[15] oft besser lesbar als eine Schrift ohne Serifen. Am Bildschirm gilt dies nicht.
- Für Systemausgaben serifenlose Proportionalschrift.
- Kursivschrift nicht für längere Textpassagen.
- Groß- und Kleinschreibung verwenden.
- Kodierungen, wie fett, gesperrt oder invers dargestellte Schrift, sollten als Gestaltungsmittel nur sparsam verwendet werden.
- Unterstreichungen machen Wörter in der Regel schwerer lesbar. Außerdem werden diese als „Link" interpretiert.
- Der Kontrast (Leuchtdichteverhältnis) zwischen Zeichen und Hintergrund muss ausreichend sein (› 1:3). Insbesondere bei farbiger Darstellung zu prüfen.
- Vertikal und horizontal benachbarte Zeichen dürfen sich einschließlich ihrer Ober- und Unterlängen nicht berühren.

Gibt es Beschwerden vieler Nutzer, sollte man die Schriftdarstellung exakt prüfen. Hierzu gibt es in der DIN EN ISO 9241-3 detaillierte Festlegungen.

[13] In Hinblick auf die demografische Entwicklung (immer mehr ältere Arbeitnehmer) ist die Verwendung größerer Schriften zu empfehlen.

[14] DIN EN ISDIN EN ISO 9241-3 Anforderungen an visuelle Anzeigen

[15] Serifen = kleine abschließende Querstriche am oberen oder unteren Ende von Buchstaben

Verwechslungsgefahr
o O 0
U V
S 5
L i 1 l
g p 9
B ß 8

Anhand dieser Zeichen kann die gute Lesbarkeit einer Schrift geprüft werden.

Dies ist eine Serifenschrift

Dies ist eine serifenlose Schrift

Kursive Schrift ist nicht gut lesbar

Computer-Schreibschriften sind zu vermeiden

SCHRIFT IN GROSSBUCHSTABEN
WIRD DEUTLICH LANGSAMER VERARBEITET ALS
Schrift mit großen und kleinen Buchstaben

Eine gute Lesbarkeit ist wichtig!

Die Lesbarkeit einer Schrift hängt nicht allein von der Größe der dargestellten Zeichen ab.
Es handelt sich vielmehr um ein Zusammenwirken verschiedener Faktoren:

- Sehabstand
- Licht- und Kontrastverhältnisse im Raum
- Schriftart (Arial, Times New Roman, MetaPlus Roman...)
- Schriftstil (fett, kursiv, shadow...)
- Pixelgröße bzw. Pixelabstand (Pitch)
- Zeichengröße (absolut bzw. in Pixeln)
- (innerer) Zeichenkontrast

Wenn diese Bedingungen aus ergonomischer Sicht in Ordnung sind, werden für verschiedene Sehabstände folgende Zeichengrößen empfohlen:

Sehabstand (mm)	Empfohlene Zeichenhöhe (mm)	Bildschirmdiagonale (CRT) (Zoll/cm)				
		15/38	17/43	19/48	21/53	› 21/53
500	3,2 bis 4,1	•	•	•	•	•
	4,2 bis 4,5		•	•	•	•
600	3,9 bis 4,1	•	•	•	•	•
	4,2 bis 5,2		•	•	•	•
	5,3 bis 5,5			•	•	•
700	4,5 bis 5,2		•	•	•	•
	5,3 bis 6,0			•	•	•
	6,1 bis 6,4				•	•
800	5,2		•	•	•	•
	5,3 bis 6,0			•	•	•
	6,1 bis 6,8				•	•
	6,9 bis 7,3					•

Bildschirmgröße für eine Textzeile in Abhängigkeit von Sehabstand und Zeichenhöhe (eine Textzeile = mind. 80 Zeichen)

Bildschirmgröße für eine Textzeile (mind. 80 Zeichen) in Abhängigkeit von Sehabstand und Zeichenhöhe (Textverarbeitungsprogramm Microsoft Word, Zeichensatz Arial 11, große Schriftarten). Für LCD-Anzeigen ergeben sich ca. 2'' kleinere Bildschirmdiagonalen.
Beispiel: Bei 600 mm Sehabstand beträgt die empfohlene Zeichenhöhe 3,9 mm bis 5,5 mm. Um 80 Zeichen pro Zeile darstellen zu können, benötigt man zwischen 3,9 mm und 4,1 mm Zeichenhöhe einen CRT-Bildschirm mit einer Diagonalen von mindestens 15''; zwischen 4,2 mm und 5,2 mm Zeichenhöhe benötigt man eine Diagonale von mindestens 17'' und ab 5,3 mm eine Diagonale von mindestens 19''.

Quelle: Bildschirm- und Büroarbeitsplätze, 2005, VBG, BGI 650 (= GUV-I 650)

Farben

Farben sind ein vertrautes Mittel, um Objekte aller Art ästhetischer zu gestalten und damit die Motivation und das psychische Wohlbefinden der Menschen, die mit diesen Objekten konfrontiert sind, zu erhöhen. Farbe ist jedoch kein objektiver Faktor, sondern eine subjektive Wahrnehmung, die von verschiedenen physikalischen und physiologischen Bedingungen abhängt und daher individuellen Schwankungen (bis hin zur Farbfehlsichtigkeit) unterliegt, so dass es sehr schwierig ist, Regeln zu definieren.
Es ist ohnehin empfehlenswert, zuerst einen Entwurf für Bildschirmmasken in Graustufen zu erstellen, der dann bei Bedarf farblich optimiert werden kann.

Der Einsatz von Farbe wird hier unter Wahrnehmungs- und Kodierungsaspekten betrachtet, ästhetische Gesichtspunkte werden hier nur am Rande erwähnt.

Folgende Regeln sind bei der Farbgestaltung am Bildschirm zu beachten:

- Die Farbgebung ist der letzte Schritt bei der Gestaltung der Benutzungsoberfläche, d.h. erst werden alle anderen Gestaltungsmöglichkeiten ausgeschöpft. So wenig Farbe wie möglich, soviel Farbe wie nötig.

- Es sollten zur sicheren Unterscheidbarkeit maximal schwarz/weiß und 6 Farben gleichzeitig verwendet werden.

- Die Zahl der Farben kann weiter reduziert werden, wenn Beziehungen zwischen zusammengehörenden Informationen durch verschiedene Helligkeits- oder Sättigungsstufen einer Farbe (z. B. hellblau, dunkelblau) dargestellt werden.
  Beispiel: Ein dunkles Grün wird für ein inaktiviertes Feld benutzt. Die Aktivierung dieses Feldes wird in hellerem Grün gezeigt.

- Für Text- oder Formularbearbeitung sollte prinzipiell die Positivdarstellung (dunkle Zeichen vor hellem Grund) verwendet werden. Zur Kennzeichnung von besonderen Feldern (z. B. Pflichtfeldern) sollten Pastellfarben verwendet werden. Zur Strukturierung von Formularblöcken bieten sich Graustufen an.

- Es sollte ein relativ geringer Helligkeitskontrast (= Leuchtdichtekontrast) angestrebt werden. Besonders bei größeren Flächen auf dem Bildschirm sollen Farben mit ähnlicher Helligkeit verwendet werden, da diese harmonisch auf den Menschen wirken. Als angenehm werden helle Farben für große Flächen und dunkle Farben für kleine Flächen empfunden.

- Für Farbkombinationen gilt die Faustregel: Falls eine Komponente (z. B. der Hintergrund) ungesättigt ist, sollte die andere gesättigt gestaltet werden.







Ergonomisch kritische Farbkombination *Ergonomisch bessere Farbkombination*

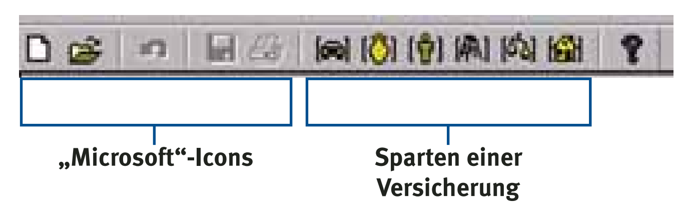
- Der Hintergrund von Fensterobjekten sollte nicht bunt sein. Ein Grauton, den Benutzer in seiner Helligkeit in der Mitte zwischen Schwarz und Weiß empfinden, ist im Normalfall als Hintergrundfarbe am besten geeignet.
- Die Kodierung mit Farbe muss konsistent sein, d. h., es dürfen nie zwei Farben dieselbe Bedeutung haben und jede Farbe muss stets in derselben Bedeutung verwendet werden.
- Standardmäßig sind die vom Betriebssystem vorgegebenen Farben zu verwenden. Eine Anpassung an individuelle Bedürfnisse ist nur in begründeten Ausnahmefällen sinnvoll.
- Ist Farbe zur Kodierung notwendig, ist ein Kodierungsplan zu erstellen.
- Die gleichzeitige und großflächige Verwendung von gesättigtem Blau ins Rot ist zu vermeiden.

Rot	„Alarm“ blinkend, falls sofortige Reaktion erforderlich
Gelb	„Vorsicht“ Grenzsituation, Vorsicht geboten
Grün	„in Ordnung“ volle Funktionsfähigkeit, Zustand korrekt
Weiss	„unbestimmt“ nicht definierbarer Zustand, Übergang, Darstellung von Alternativen
Blau	„Hilfsfarbe“ am besten vermeiden

Bedeutung von Farben

3.6 Gestaltung von Icons[16]





Individuelle Icon-Gestaltung angepasst an Standard-Icons

Gerade bei der Gestaltung von Icons für Web-Software spielt das Design und/oder Corporate Design eine Rolle. Diese Design-Anforderungen allein reichen jedoch nicht aus. Wenn es Fehlinterpretationen der Benutzer gibt, nützt kein grafisch einwandfreies (= ästhetisch) gestaltetes Icon. Es reicht auch nicht, wenn das Icon gefällt, aus ergonomischer Sicht muss das Icon nachweislich funktionieren, d.h., es muss ohne Erklärung verstanden werden und unverwechselbar sein.

Neben der Verwendung von Microsoft-Standard-Icons, kann es erforderlich sein, eigene Icons zu entwickeln, z. B. für die verschiedenen Sparten einer Versicherung.

Regeln für die Gestaltung sind:

- Ein Icon sollte ein Referenzobjekt oder charakteristische Merkmale in stark abstrahierter visueller Form darstellen.

Um die richtige Lösung zu finden, muss man potentielle Benutzer befragen. Dabei geht es nicht um Geschmack, sondern darum, was Benutzer mit dem Icon assoziieren.

- Benutzer sollen von der Art der Darstellung die Bedeutung erkennen können („Lupe“ für Suchen).
- Möglichst geschlossene Figuren verwenden.
- Dunkle Figuren vor hellem Grund bevorzugen.
- Es sollte eine vertikale Hauptachse geben, um die herum das Icon symmetrisch gestaltet ist.
- Einheitliche Größe und Proportionen bei mehrfach vorkommenden Elementen.
- Details der Icons sollten auch noch aus einem Sehabstand von 50 cm erkennbar sein.
- Icons sollten kultur- und bildungsneutral sein.
- Eine inflationäre Verwendung von Icons führt zu einer unübersichtlichen Oberfläche und behindert den schnellen Zugriff auf die Funktionalitäten.
- Die Icons in der Toolbar unter dem Menü dienen dem schnellen Zugriff auf häufig benutzte Funktionen aus dem Menü. Welche dies sind, muss mit Benutzern ermittelt werden. Das bedeutet darüber hinaus, dass es kein Icon mit einer Funktion geben darf, die nicht im Menü zu finden ist. Diese Icons sollten mit einem Text (Quickinfo = tooltip) versehen werden, der erscheint, wenn man mit der Maus über das Icon fährt (siehe Abbildung links oben).



System-Icons



Anwendungs-Icons



Geräte-Icons



Container-Icons



Daten-Icons

[16] Icons (= Sinnbild) sind dem Objekt, das sie symbolisieren, ähnlich.

3.7 Gestaltung von Texten

Regeln zur Gestaltung von Texten sind auf den unterschiedlichsten Ebenen zu beachten, z. B. bei der Wahl der Feldbezeichner, bei der Formulierung von Menüeinträgen oder bei der Formulierung der verschiedenen Meldungstexte.

- Das Vokabular soll sich an der Aufgabe des Benutzers orientieren und für diesen verständlich sein.

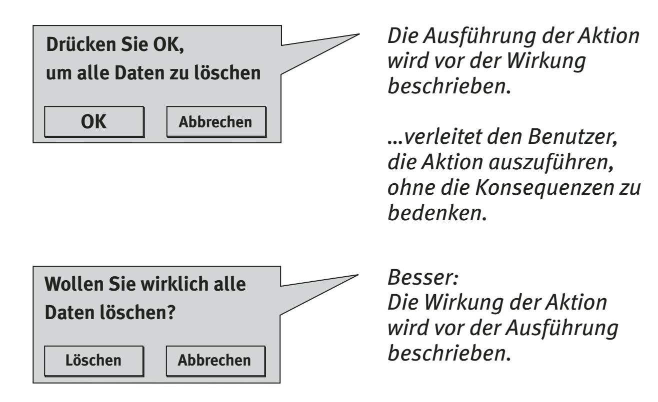
- Möglichst deutsche Begriffe verwenden.

- Nicht ausssagekräftige Wortverlängerungen (Suffixe) vermeiden. Das Suffix „-nummer“ oder „-name“ ist oft überflüssig. Suffixe wie „-schlüssel“ und „-kennzeichen“ sollten nur dann benutzt werden, wenn sie für das Verständnis des Feldes notwendig sind.

- Keine Abkürzungen verwenden, wenn genügend Platz zur Verfügung steht oder Benutzer nur selten auf diese Maske kommen. Gängige Abkürzungen wie PLZ oder BLZ sollten jedoch verwendet werden. Möglichst einfache Abkürzungsregeln verwenden (z. B. erste 3 Buchstaben, erste 3 Konsonanten, ansonsten sollten Abkürzungen sprechbar sein).

- Wenn Feldbezeichner aus zwei oder mehr Wörtern bestehen, soll der informationstragende Teil dieser Bezeichnung vorne stehen.
  schlecht: Bestand alt
  besser: alter Bestand

- Texte in Frageform sollten bejahend formuliert sein.
  schlecht: Möchten Sie die Daten nicht sichern?
  besser: Möchten Sie die Daten sichern?



Rückmeldungstexte

- Verwenden von kurzen, einfachen Sätzen. Die Redeform (aktiv/passiv) sollte dem Sprachgebrauch entsprechen.

- Wenn man in Kurzform formuliert, wählt man die Reihenfolge Substantiv-Verb und lässt den Artikel weg. Es ist zu vermeiden, Verben zu substantivieren.

- Meldungen sind so präzise wie möglich zu formulieren. Sie sollen die aufgetretene Situation möglichst eindeutig beschreiben. Meldungen, die sich auf technische Funktionsweisen des Systems beziehen, sind zu vermeiden.

- Grundsätzlich sollten Nachrichten spezifische Informationen beinhalten (nicht nur „Eingabefehler“).

- Die syntaktische Struktur soll klar und einfach sein und aus vollständigen Satzteilen bestehen. Artikel können weggelassen werden; ein Verb (mindestens: ist, sind) sollte aber immer vorhanden sein; als Zeit ist nach Möglichkeit die Gegenwartsform zu wählen.

- Es sollte zuerst das Ergebnis angegeben sein, dann wird beschrieben, wie die Aktion ausgeführt werden soll, z. B. „Zum Löschen von... drücken Sie...“

- Die Formulierungen sollten benutzerbezogen sein, z. B. „Mit der OK-Taste speichern Sie Ihre Änderungen“, statt „Das System speichert...“.

- Formulieren, was zu tun ist, statt, was zu vermeiden ist.

- Der Ton ist freundlich und positiv zu halten und darf nicht belehrend klingen. Nichtssagende oder belehrende Worte wie „dürfen“, „müssen“, „sollen“, „gehören“, „falsch“, „unzulässig“, „unvollständig“, „nicht erlaubt“ oder ähnliches sind deshalb nicht zu verwenden.

Negativbeispiel	**Positivbeispiel**
Datei drucken	Drucke Datei
Eingabe der Kontonummer	Kontonummer eingeben

3.8 Tastatursteuerung

Da viele Benutzer in der Einarbeitungsphase für ein neues Programm mit der Maus arbeiten, wird die Tastatursteuerung oft vernachlässigt. Es stellt sich aber heraus, dass Benutzer nach einiger Zeit der Verrichtung von Routinetätigkeiten mit den Händen bevorzugt auf der Tastatur bleiben. Dieses gilt es auch aus physiologischen Gründen zu unterstützen („Mausarm“).
Deshalb sollten Benutzer – soweit möglich – Programme komplett über die Tastatur steuern bzw. bedienen können (DIN EN ISO 9241-16). Dies bezieht sich sowohl auf die Navigation als auch auf das Auslösen von Aktionen, die beispielsweise über Menüs oder Push Buttons aktiviert werden können.
Hierfür gibt es allgemein gültige Regeln. In der Praxis wird hier jedoch gelegentlich verwechselt, dass es im einen Fall um die prinzipielle Tastaturbedienung geht und im anderen Fall um den Schnellzugriff für den geübten Benutzer.

Mnemonics für Tastaturbedienung

Accelerator für Schnellzugriff

Möglichkeiten der Tastaturbedienung

3.8.1 Zugriffstasten (Mnemonics)

Um dem Benutzer ein tastaturgestütztes Arbeiten zu ermöglichen, werden u. a. Mnemonics (Mneme = Gedächtnis) definiert. Dies sind leicht merkbare Abkürzungen, um bestimmte Aktionen per Tastaturbefehl auszulösen. Beispiel: Um eine Datei über die Tastatur zu speichern, kommt man mit Alt-D in das Dateimenü. In diesem löst man dann durch Eingabe der Tastenkombination Strg+S das Speichern der aktuellen Datei aus. Mnemonics werden in der Reihenfolge Menü, Drucktasten, Register, Feldbezeichner vergeben.

Es muss sichergestellt sein, dass die festgelegten Mnemonics im aktuellen Kontext eindeutig gewählt sind. D. h., für zwei Objekte darf parallel bzw. zur gleichen Zeit nicht dasselbe Mnemonic definiert werden.

Den Hauptbegriffen des Menüs sind applikationsweit eindeutige Mnemonics zuzuordnen. Die Mnemonics werden nach folgenden Regeln festgelegt:

- Bei gleichlautenden Oberbegriffen zu Microsoft-Applikationen wie beispielsweise MS-Word werden die von Microsoft definierten Mnemonics verwendet.

Für applikationsspezifisch gewählte Oberbegriffe werden die Mnemonics systematisch wie folgt definiert:

- Es wird der erste Buchstabe gewählt, der noch nicht vergeben ist, beginnend beim Anfangsbuchstaben.
- Das definierte Mnemonic wird unterstrichen und so dem Benutzer visualisiert.
- Die Tastatursteuerung erfolgt gemäß des Standards für Applikationen in Windows-Umgebungen mit der Tastaturkombination ‹Alt› ‹Mnemonic›.
- Eine Differenzierung zwischen Groß- und Kleinbuchstaben erfolgt nicht.

Mnemonics (anwendungseinheitlich)	**Accelerator Keys (anwendungs-übergreifend)**
für alle Menüeinträge, Tool Bar, Push Buttons	nur in Pull Down/ Kaskadenmenüs
z. B. Drucken	z. B. Strg + C
Reihenfolge: Menü, Drucktasten, Tab Control, Feldbezeichner	rechts im Menü darstellen
über Groß- und Kleinbuchstaben erreichbar	es gibt auch anwendungsspezifische Accelerator Keys

Tastenkombination für grundsätzliche Tastaturbedienbarkeit

3.8.2 Schnelltasten (Accelerator Keys)

Ebenfalls applikationsweit und somit einheitlich werden hinter bestimmten Tastaturkombinationen allgemeine Funktionen definiert. Auf diese Weise wird das tastaturgestützte Arbeiten von Benutzern optimal ergänzt. Diese so genannten Accelerator Keys (Schnelltaste = Beschleunigen) dienen dem schnellen Zugriff auf häufig benutzte Aktionen. So kann der Befehl „Datei speichern“ durch die Tastenkombination Strg + S ausgelöst werden. Accelerator sind im Menü neben der Bezeichnung aufgeführt.
Auch hier gilt, dass generell die im Windows-Umfeld üblichen Tastaturkombinationen individuellen Festlegungen vorgezogen werden.

Tastenkombination	**Aktion**
Strg ⇠S⇢	Speichern
Strg ⇠X⇢	Ausschneiden
Strg ⇠C⇢	Kopieren
Strg ⇠V⇢	Einfügen
F1	Aufruf der Hilfe
F4	Wiederholen des letzten Befehls
Tab	Sprung zum nächsten GUI-Element
Shift (Hochstelltaste) + Tab	Rücksprung zum vorherigen GUI-Element

Tastenkombinationen für den schnellen Zugriff



4 Das Software-Produkt: Dialoggestaltung

4.1 Softwaregestaltung ist Arbeitsgestaltung (DIN EN ISO 9241-2)

Unterscheidet man bei der Gestaltung einer Software in ”look“ und ”feel“, geht es in diesem Kapitel um das ”feel“, d. h., wie fühlt sich der Umgang mit der Software an, wie werden Handlungsinteraktionen unterstützt.
Software wird anhand von Geschäftsprozessen entwickelt. Geschäftsprozesse sind Abläufe innerhalb einer Organisation aus Sicht des Managements. Aus der Sicht der Beschäftigten stellen sich diese als Szenarien von Arbeitsabläufen da. Wichtig ist bei jeder Softwareentwicklung beide Sichtweisen zu berücksichtigen.

Die DIN EN ISO 9241 bezieht sich auch auf die Gestaltung von Software. Da Softwaregestaltung immer auch Arbeitsgestaltung bedeutet, finden sich im Teil 2 der Normenreihe Aussagen zur ergonomischen Arbeitsgestaltung. Arbeits- und Aufgabengestaltung ist dort als wichtige Voraussetzung für den Software-Entwurf beschrieben.
In der Arbeitsgestaltung wird unterschieden zwischen der Person, die arbeitet, der eigentlichen Arbeitsaufgabe und dem Umfeld, in dem diese Arbeit stattfindet.
Hintergrund ist, dass eine Arbeitsaufgabe, die z. B. aus ständig monoton wiederkehrenden Tätigkeiten besteht, nicht allein durch eine Software verbessert wird. Unzufriedenheit von Benutzern kann demnach nicht nur aus Software-Produktmerkmalen, sondern auch aus Aufgabenmerkmalen resultieren. Deshalb sollte eine Systementwicklung zur Realisierung der hier beschriebenen Ziele beitragen und umgekehrt sollten erst aus den gut gestalteten Arbeitsaufgaben Anforderungen an die Software abgeleitet werden.

Arbeitsaufgaben sind dann ergonomisch gut gestaltet, wenn[1]

- die Erfahrungen und Fähigkeiten der Benutzer berücksichtigt werden,
- sie die Entfaltung einer angemessenen Mischung unterschiedlicher Fertigkeiten, Fähigkeiten und Tätigkeiten ermöglichen,
- die durchgeführten Arbeitsaufgaben als ganze Arbeitseinheiten erkennbar sind und nicht nur als Bruchstücke,
- die durchgeführten Arbeitsaufgaben einen wesentlichen Beitrag zur Funktion des gesamten Systems leisten, der für den Benutzer erkennbar ist,
- sie dem Benutzer einen angemessenen Handlungsspielraum bei der Entscheidung über die Reihenfolge, das Arbeitstempo und die Vorgehensweise ermöglichen,
- ausreichende Rückmeldungen über die Durchführung der Arbeitsaufgabe vom System an den Benutzer erfolgen,
- Möglichkeiten zur Entwicklung vorhandener und zur Aneignung neuer Fertigkeiten im Rahmen der Aufgabenstellung gegeben sind.

Weiter heißt es:

Arbeitsaufgaben sind so zu gestalten, dass

- die Durchführung der Arbeitsaufgabe erleichtert wird,
- die Gesundheit und Sicherheit der Mitarbeiter geschützt werden,
- das individuelle Wohlbefinden gefördert wird,
- die individuellen Fertigkeiten und Fähigkeiten im Rahmen der Aufgabenstellung weiterentwickelt werden können.

Dabei sind insbesondere zu vermeiden:

- Über- und Unterforderung
- unangemessene Häufigkeit der Wiederholung immer gleicher Arbeitsvorgänge
- unangemessener Zeitdruck
- isoliertes Arbeiten ohne die Möglichkeit zu sozialen Kontakten

Diesen Forderungen liegen Erkenntnisse aus dem Bereich der Arbeitswissenschaften[2] zugrunde.
Eine der zugrunde liegenden Annahmen ist die Theorie von der menschlichen Handlungsregulation. Kurz zusammengefasst besagt diese, dass Menschen unter normalen Umständen[3] ihre Handlungen zur Aufgabenbearbeitung in folgenden Schritten planen:

- Handlungsziel bilden
- Handlung/Aktion zur Zielerreichung auswählen
- Handlung ausführen
- Handlungsergebnis bewerten
- ggf. neues Ziel bilden

---

[1] DIN 29241-2

[2] Arbeitswissenschaften ist ein Sammelbegriff für Wissenschaftsteilgebiete, die sich mit der Analyse, Bewertung und Gestaltung menschlicher Arbeit befassen.

[3] Experten in sicherheitskritischen Systemen handeln nachweislich anders. Dieser Bereich wird hier jedoch nicht behandelt.

Entscheidende Stärke des menschlichen Denkens ist die rasche Ein- und Umstellfähigkeit, die es erlaubt, auch unerwartete Situationen mit ad hoc entwickelten Handlungsstrategien zu meistern. In sicherheitskritischen Systemen etwa sollte der dreistufige Vorgang der menschlichen Reaktion auf auftretende Ereignisse unterstützt werden. Zuerst nimmt der Mensch wahr, dann entscheidet er im Kopf und dann erst führt er Handlungen aus.

Bei der Software-Gestaltung geht es nun darum, diese Handlungsreaktion zu unterstützen bzw. nicht zu behindern.

Bei der ergonomischen Arbeitsaufgabengestaltung unterscheidet man zwischen Regulationserfordernissen und Regulationsbehinderungen. Das Regulieren einer Arbeitsaufgabe erfordert einerseits bestimmte Handlungsspielräume und kann andererseits durch regelrechte Behinderungen (schlechte Unterscheidbarkeit von Informationen) oder Über-/Unterforderung (z. B. hohe Anforderungen an Kurz- und Langzeitgedächtnis durch unangemessene Menüstruktur) erschwert werden. In der Bildschirmarbeitsverordnung wird gefordert, dass diese aufgabenbezogenen psychischen Belastungen durch angemessene Dialoggestaltung zu vermeiden sind.

Arbeitswissenschaftler haben definiert, dass Behinderungen des Arbeitshandelns vorliegen, wenn die Aufgabenerfüllung durch betrieblich festgelegte oder organisatorisch/technisch (Software!) entstandene Bedingungen erschwert, behindert oder blockiert wird. Diese Behinderungen sind, wenn sie von den Beschäftigten nicht grundsätzlich beseitigt werden können, objektiv belastend, da sie zusätzlichen Aufwand verlangen.

Arbeitsbedingungen werden als psychisch belastend verstanden, wenn

- sie den/die Arbeitende(n) bei der Aufgabenerledigung behindern und
- ihm dadurch überflüssigerweise zusätzlichen Handlungsaufwand abfordern[4].

Nach einer Untersuchung[5] verbringen Benutzer 6–7 Stunden pro Woche mit der Abarbeitung von derartigen Arbeitsbehinderungen (Regulationsbehinderungen).

Belastungen bei der Arbeit mit Software können sein:

Informatorische Erschwerungen

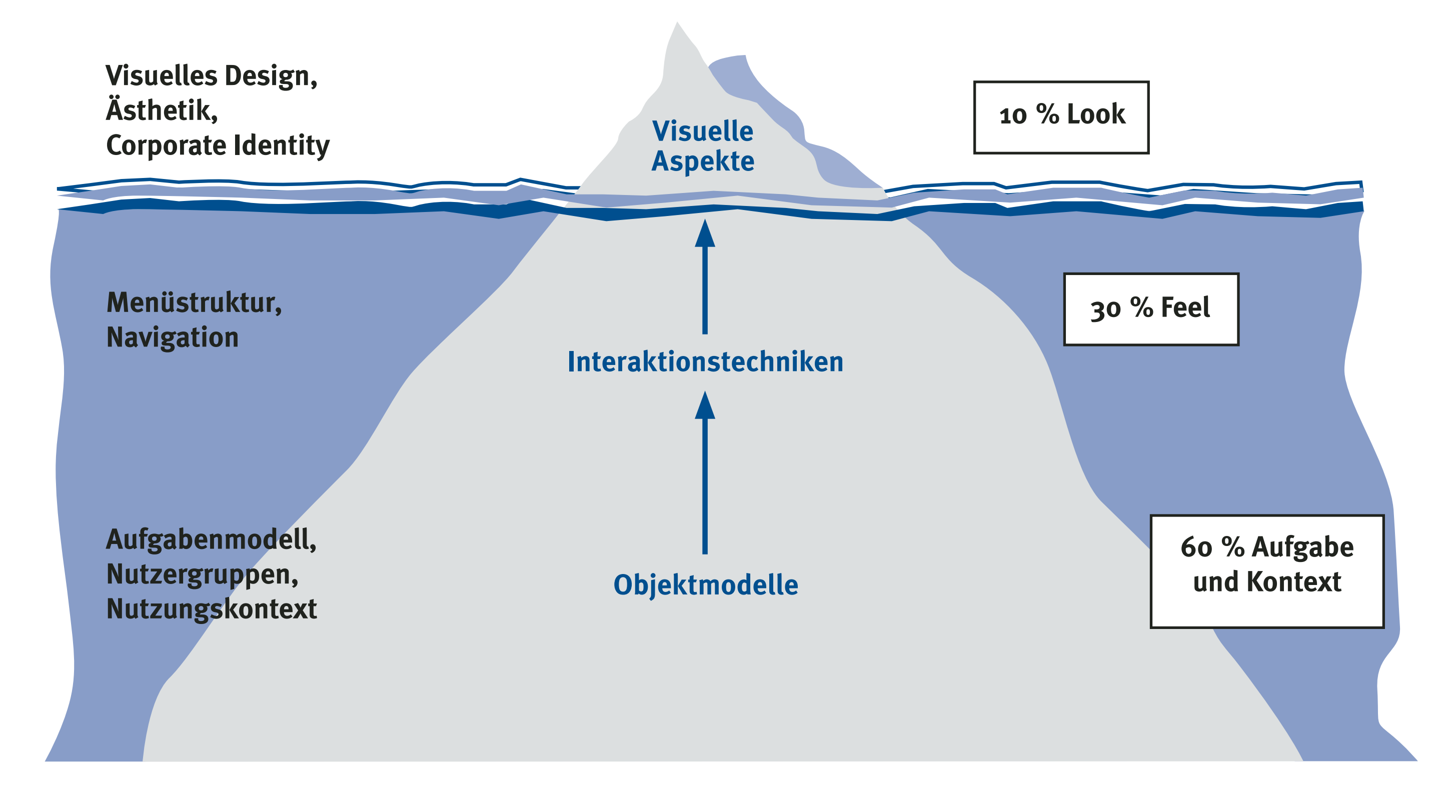
- unübersichtliche, unstrukturierte und informationell überladene Masken
- uneinheitliche Oberfläche unterschiedlicher Anwendungen
- die Verteilung von bei einem Arbeitsschritt benötigten Informationen auf unterschiedliche Masken/Anwendungen
- fehlende, falsche oder unverständliche Fehlermeldungen
- fehlerhafte oder nicht aktuelle Daten
- unklare Systemzustände bzw. Systemmeldungen

Motorische Erschwerungen

- fehlende Möglichkeiten, Eingaben zu korrigieren
- fehlende Möglichkeiten der Zwischenspeicherung
- fehlende Möglichkeit des unmittelbaren Zugriffs auf häufig benötigte Masken
- unangemessene automatische Cursorpositionierung
- Systemvorgaben, die bei der Aufgabenerledigung nicht angemessen sind
- umständlicher Wechsel zwischen unterschiedlichen Anwendungen

4 Ulich, E., Arbeitspsychologie, 1994

5 Software-Ergonomie 91, Hrsg. Ackermann, D., Ulich, E. 1991, S. 370



Hohe Nutzungsqualität zeigt sich wenig an der Oberfläche einer Software

Quelle: in Anlehnung an IBM

http://www-106.ibm.com/developerworks/library/w-berry/

Weil sich hohe Nutzungsqualität kaum an der Oberfläche einer Software zeigt, haben die im folgenden Abschnit 4.2 beschriebenen Kriterien einen hohen Stellenwert.

4.2 Ergonomische Dialogkriterien (DIN EN ISO 9241-10)

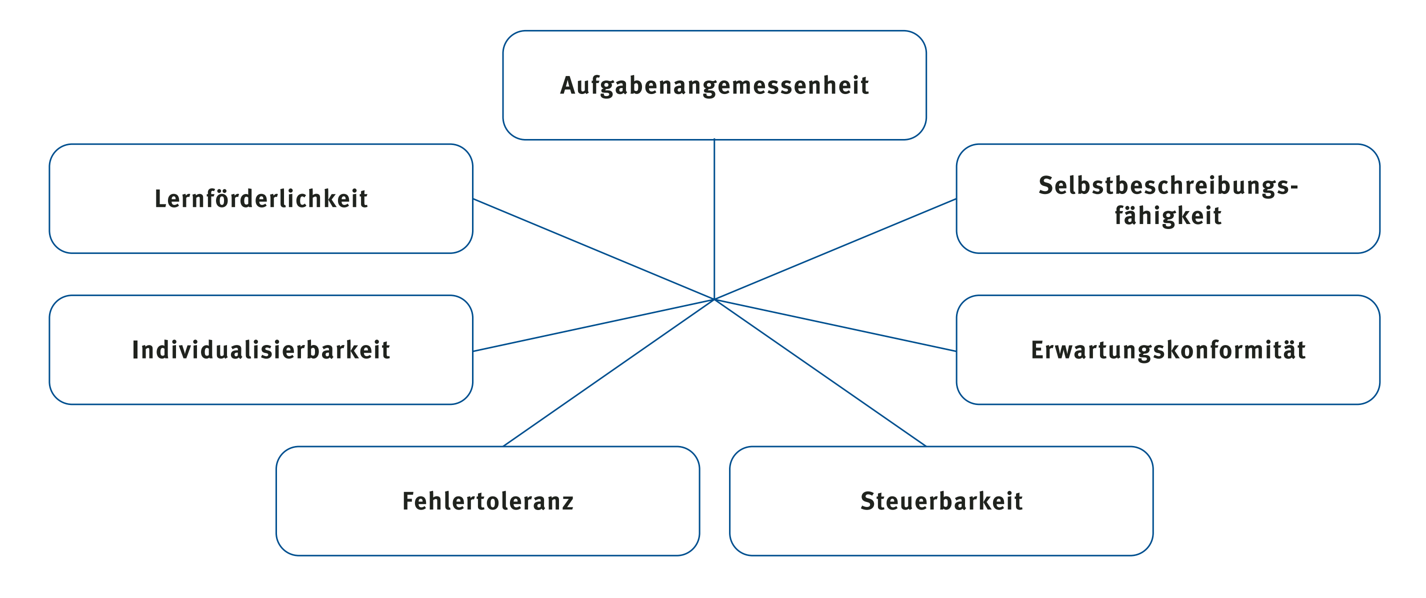
Die Kunden wollen keine Bohrer, sondern Löcher.

Als Dialog wird die Interaktion zwischen einem Benutzer und einem Dialogsystem verstanden, um ein bestimmtes Ziel zu erreichen. Die Dialogführung kann technisch realisiert sein als Formulardialog, über Menüsteuerung, direkte Manipulation oder auch über eine Kommandosprache. In der Praxis hat man es im allgemeinen mit Mischformen zu tun. Die folgenden Grundsätze ergonomischer Dialoggestaltung gelten technik-unabhängig und sind deshalb für den Einzelfall zu konkretisieren.

Aufgabenangemessenheit – Wenn es direkt und einfach zum Ziel geht [6]

Aufgabenangemessenheit ist dann gegeben, wenn der Benutzer durch das System bei der Bearbeitung seiner Aufgabe effizient und effektiv unterstützt wird.
Dies ist dann der Fall, wenn der Benutzer durch die Systembenutzung nicht vom eigentlichen Aufgabenziel abgelenkt wird, sondern er es mit weniger oder gleichem Aufwand erreicht und es genauso bzw. vollständiger erledigt als zuvor.
D. h., der Benutzer wird nur mit Tätigkeiten und Konzepten bei der Lösung seiner Arbeitsaufgabe konfrontiert, die in Bezug zu dieser Arbeitsaufgabe stehen und so angepasst sind, dass sie eine effektive und effiziente Lösung unterstützen.
Aufgabenangemessen heißt auch, dass die technische Unterstützung für eine Aufgabe angemessen sein muss. So ist es denkbar, dass man Teile einer Aufgabe nicht technisch unterstützt, da in diesen Aufgabenteilen der Mensch mit seinen Fähigkeiten besser entscheidet. Dies setzt eine Aufgabenanalyse voraus, in der man eben feststellt, welche Aufgabenteile besser vom PC unterstützt werden und welche beim Menschen verbleiben sollten.

So kann eine Software für die Kreditvergabe z. B. besser Berechungen anstellen, dem Menschen sollte allerdings die Entscheidung über die weichen Faktoren der Kreditwürdigkeit überlassen bleiben. Es kann sinnvoll sein, dass das System einen Vorschlag macht, dieser sollte aber dann „überschreibbar“ sein. Gerade die Fähigkeit von Menschen, mit unvorhersehbaren Situationen umzugehen, sollte nicht durch technische Einschränkungen unterbunden werden. Diese Diskussion ist besonders für Software in sicherheitskritischen Bereichen interessant.
Aufgabenangemessene Dialoge können nur über die Ermittlung von Use-Szenarien (nicht Geschäftsvorfällen) ermittelt werden (siehe Abschnitt 8.3).

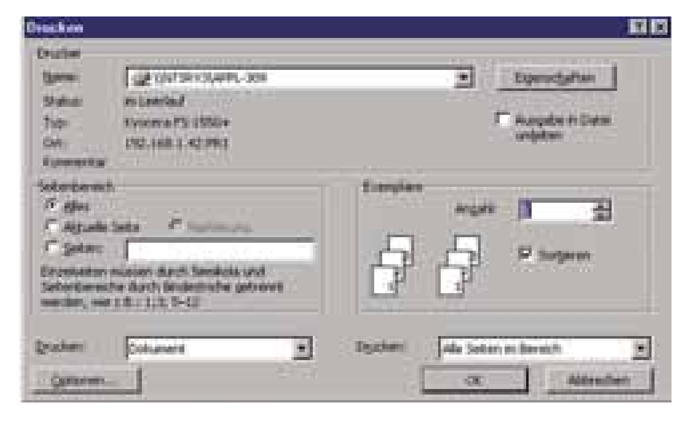


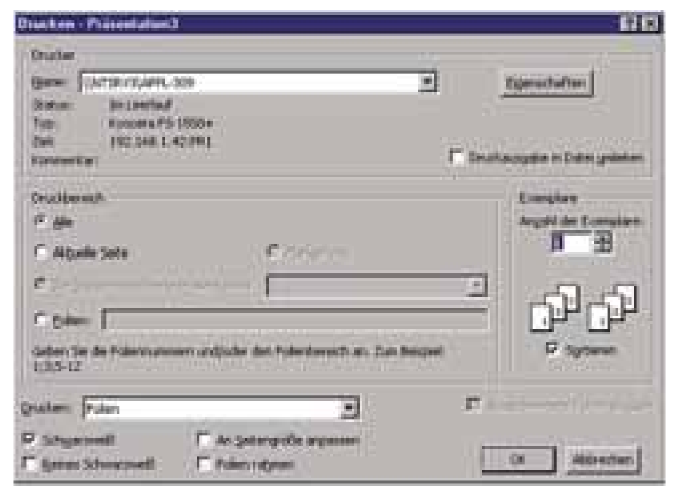
Grundsätze der Dialoggestaltung

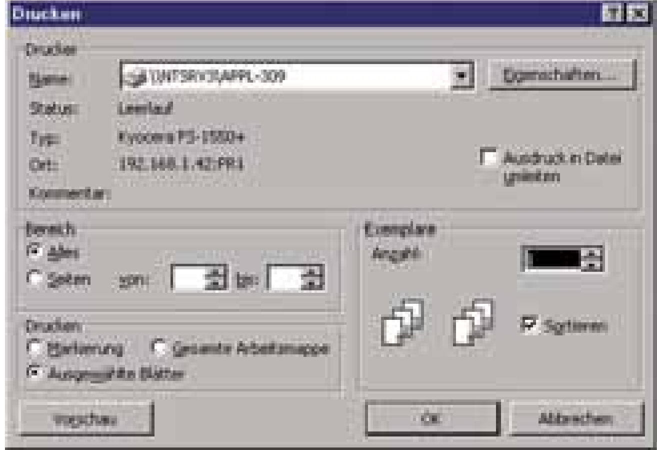
[6] Diese umgangssprachlichen Beschreibungen der Dialoggrundsätze sind von http://www.kommdesign.de

Folgende Punkte unterstützen die Aufgabenangemessenheit der Anwendung:

- Jede Benutzergruppe findet ihre Sicht der Arbeitsabläufe, d. h. Tätigkeiten und Objekte, in der Anwendung wieder.
- Arbeitsvorgänge, die der Benutzer aus seinem Verständnis heraus als zusammengehörig empfindet, spiegeln sich in der Reihenfolge der Bearbeitungsschritte wider.
- Mediensprünge (z. B. Zwischenrechnungen auf Papier) sind nicht notwendig.
- Standardwerte werden dem Benutzer als Vorgabe angeboten, können jedoch überschrieben werden.
- Auf einer (Intra-)Website sind Ansprechpartner zu finden, so dass ein Benutzer/Kunde sein Ziel, eine E-Mail persönlich zu adressieren, erreichen kann.
- In einem Formular werden keine Pflichtangaben verlangt, die mühsam auszufüllen sind und mit dem Abwickeln des relevanten Vorgangs nichts zu tun haben.
- Bei der Gestaltung von Grafiken wird auf minimale Wartezeiten (beim Bildaufbau) Wert gelegt.
- In einem Formular, das korrigiert werden muss, wird der Cursor gleich auf das zu korrigierende Feld gesetzt.
- Zwischenergebnisse einer längeren Online-Transaktion können gespeichert werden.
- Für Benutzer, die ein Angebot oft nutzen, werden Short Cuts zu den wichtigsten Seiten bereitgestellt.







Ein im wesentlichen identischer Druckdialog ist bei verschiedenen Anwendungen ähnlich dargestellt.

Selbstbeschreibungsfähigkeit – Wenn Intuition ausreicht
Eine Anwendung ist selbstbeschreibend und unterstützend, wenn jeder (Teil-)Schritt des Dialogs für den Benutzer sofort verständlich ist. Diesen Effekt kann man u.a. durch unmittelbares Rückmelden erreichen. Ebenso können notwendige Informationen auf Wunsch des Benutzers gegeben werden.

Die folgenden Punkte unterstützen die Selbstbeschreibungsfähigkeit einer Anwendung:

- Einzelne Schritte sollten zurückgenommen werden können.
- Die Ausführung von irreversiblen Schritten sollte ggf. mehrfach abgefragt werden. Dem Benutzer muss deutlich das Ausmaß der irreversiblen Veränderungen angezeigt werden (Transparenz).
- Der Benutzer muss über den Erfolg oder Misserfolg aller Aktionen informiert werden. Das bedeutet, dass (Fehler-) Meldungen explizit angezeigt werden sollen (Rückmeldung).
- Die Sprache und die Fachsprache des Benutzers sollte verwendet werden. Dies erleichtert dem Benutzer die Interpretation, und er kann sein Ziel schneller erreichen.
- Dem Benutzer sollen bei ähnlichen Aufgaben ähnliche, sinngemäße oder analoge Arbeitswege angeboten werden.
- Links sind so formuliert, dass man sicher vorhersagen kann, wohin sie führen.
- Der Umfang einer Trefferliste kann gleich am Tabellenanfang abgelesen werden.
- Eine Web-Applikation hat eine Online-Hilfe, die kontextspezifische Bedienhinweise gibt.
- Nachdem eine Anfrage an eine Datenbank gesendet wurde, erscheint eine Meldung „Anfrage wird bearbeitet, bitte warten“.



Gute Selbstbeschreibungsfähigkeit reduziert die Notwendigkeit, sich erläuternde Informationen zu holen.

Steuerbarkeit – Wenn alles unter Kontrolle ist

Eine Anwendung ist steuerbar, wenn es dem Benutzer möglich ist, die Richtung der Abfolge von Interaktionsschritten bis zum Erreichen des Aufgabenziels zu beeinflussen.

Die folgenden Punkte unterstützen die Steuerbarkeit einer Anwendung:

- Die Abfolge einzelner Arbeitsschritte ist so wenig wie möglich vorgegeben. Der Benutzer kann z. B. eine Bildschirmmaske in beliebiger Reihenfolge ausfüllen.

- Der Benutzer kann den Dialog an beliebiger Stelle unterbrechen. Der Benutzer gibt an, ob er den Dialog an der Unterbrechungsstelle fortführen oder den Dialog beenden will.

- Die Anwendung reagiert sofort auf Benutzereingaben. Beim Suchen von Datensätzen wird der erste Datensatz sofort angezeigt, die Suche weiterer Datensätze erfolgt im Hintergrund.

- Eine Tabelle hat Buttons, mit deren Hilfe die Informationen spaltenweise sortiert werden können.

- Eine Suchmaschine bietet die Möglichkeit, die Zahl der auf einer Seite anzuzeigenden Treffer einzustellen.

- Ein Tool ermöglicht es den Benutzern, einen Dateidownload zu unterbrechen und später fortzusetzen.

- Umfangreiche Grafiken werden als „Thumbnails“[7] dargestellt, die bei Bedarf vom Benutzer vergrößert werden können.



Die Bezeichnung „OK“ ist zu vermeiden, da nicht handlungsleitend.

Erwartungskonformität – Wenn es zur Gewohnheit wird

Ein Anwendungsprogramm ist erwartungskonform, wenn es den Kenntnissen aus bisherigen Arbeitsabläufen, der Ausbildung und den Erfahrungen des Benutzers, sowie allgemeinen Konventionen entspricht.

Die folgenden Punkte unterstützen die Erwartungskonformität einer Anwendung:

- Bei ähnlichen Arbeitsaufgaben sollte der Dialog ähnlich gestaltet sein.

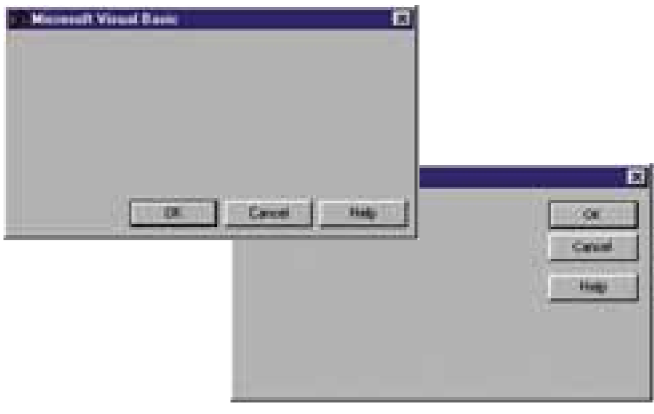
- Inhaltlich zusammengehörige Objekte sollen in Gruppen zusammengefasst dargestellt werden.

- Gleichartige Informationen sollen innerhalb einer Anwendung immer gleich bezeichnet und an den gleichen Positionen angezeigt werden.

- Push Buttons sind in den Bildschirmfenstern immer auf die gleiche Weise anzuordnen.

- Gleichartige Funktionen sind immer auf die gleiche Art und Weise zu aktivieren.

- Bei ähnlichen Aktionen sollten die Antwortzeiten ähnlich sein, bei kalkulierbaren Abweichungen sollte der Benutzer hierüber informiert werden.



Negativbeispiel (unterschiedliche Platzierung): Die Drucktasten sollten bei gleichartigen Masken immer an der gleichen Stelle angeordnet sein.

---

[7] Thumbnail: kleines Bild







Übereinstimmende Tastaturkürzel für die gleichen Funktionen in verschiedenen Programmen einer Produktfamilie.

- Jedes Objekt besitzt in allen Systemzuständen die gleichen Eigenschaften.
- Beim Drücken der Tabulator-Taste springt der Cursor auf das nächste Eingabefeld.
- Der Link zur Startseite ist unter dem Firmenlogo oben links platziert.
- Der „Warenkorb“ in einem Online-Shop heißt immer und in allen Zusammenhängen „Warenkorb“.
- Unterstrichene Wörter sind immer Hypertext-Links.

1	2	3	23,1%
11,5 %	6	7	8
17,3%	10	11	12
13,5%	14	15	16
17	18	19	20

Positionierungserwartung eines Hilfe-Links auf einer Webseite

Quelle: e-Result, GmbH

Interessant ist, dass ein Viertel der Testpersonen den Link dort (oben rechts) platzieren, wo bei vielen Softwareprodukten (z. B. MS-Office-Paket) ebenfalls eine Hilfe angeboten wird.

Fehlerrobustheit – Wenn unliebsame Überraschungen ausbleiben

Ein Dialog ist fehlerrobust, wenn trotz erkennbar fehlerhafter Eingaben das beabsichtigte Arbeitsergebnis mit keinem oder nur minimalem Korrekturaufwand erreicht wird.

Die folgenden Punkte unterstützen die Fehlerrobustheit einer Anwendung:

- Durchgeführte Aktionen können, sofern notwendig, jederzeit zurückgenommen werden.[8]
- Eingaben dürfen nicht zu undefinierten Systemzuständen führen.
- In der Regel werden Fehlermeldungen sofort ausgegeben.
- Fehlermeldungen beinhalten Hinweise zum Ort des Fehlers, der Fehlerursache und den Korrekturmöglichkeiten.
- Fehlermeldungen sind verständlich formuliert und einheitlich strukturiert.
- Ist ein Fehler oder ein Problem auf verschiedene Weise behebbar, so werden dem Benutzer die Korrekturalternativen oder die Möglichkeit eines sinnvollen Neuanfangs angeboten.
- Vor der Durchführung kritischer Aktionen (Datenverlust) sollte es für den Benutzer eine Rückfrage geben.
- Fehlermeldungen werden nicht technisch verklausuliert oder als Nummer angezeigt, sondern in der Sprache der Benutzer formuliert.

Erlernbarkeit – Wenn man schlauer wird

Das Kriterium der Erlernbarkeit bezieht sich auf die Zeit, die ein Benutzer benötigt, um die Anwendung (nicht die damit unterstützte Aufgabe) zu beherrschen. Je schneller dies geschieht, desto besser erlernbar ist das System.
Die folgenden Punkte unterstützen die Erlernbarkeit einer Anwendung:

- Die Anwendung ist so aufgebaut, dass der Benutzer die Struktur logisch nachvollziehen kann. Die Logik von gewohnten Arbeitsabläufen spiegelt sich in der Benutzungsoberfläche wider.
- Die angebotene Information beschränkt sich auf die Menge von Informationen, die zur Erledigung der einzelnen Aufgaben benötigt wird.
- Tastenkombinationen, Abkürzungen oder Tastenkürzel sind aufgrund der zugrundeliegenden Logik leicht zu verstehen und so leicht zu behalten.
- Aktionen, zu denen es eine Umkehraktion gibt, sollten so auszulösen sein, dass der Benutzer eine generelle Regel zwischen Aktion und Umkehraktion erkennen kann.
- Die verwendeten Begriffe entsprechen der Sprache bzw. der Fachsprache des Benutzers.
- In einer „Guided Tour“ werden die Benutzer mit besonderen Tricks in der Bedienung einer Applikation vertraut gemacht.
- In einer Site-Map kann man sich ansehen, nach welcher Logik eine Web-Site strukturiert ist.



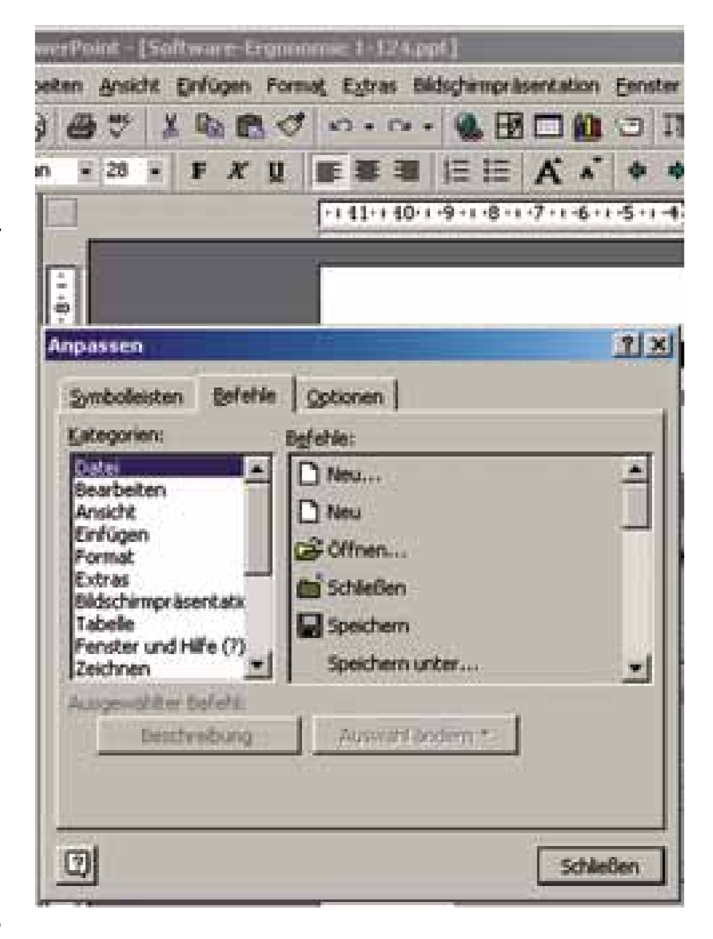
Die Funktion „Rückgängig“ steht zur Verfügung. Tastenkombination, Abkürzungen o. ä. sind leicht zu merken.

[8] Dies gilt z. B. für das Formatieren von Textabschnitten. Das Rückgängigmachen ist jedoch nicht in jedem Fall möglich bzw. auch nicht sinnvoll. Eine Sicherheitsabfrage für den Benutzer ist dann erforderlich.

Individualisierbarkeit – Wenn jeder arbeitet, wie es ihm beliebt

Anwendungen sind individualisierbar, wenn es dem Benutzer möglich ist, sie an seine speziellen Anforderungen und seinen Wissensstand anzupassen. Anwendungen werden meist von Benutzern mit unterschiedlichem Kenntnisstand genutzt. Ein neuer Benutzer wird, je länger er mit der Anwendung arbeitet, seine Kenntnisse bezüglich der Anwendung erweitern.

- Für ungeübte Benutzer und Gelegenheitsbenutzer ist eine ausführliche Benutzerführung gefordert. Erfahrene Benutzer sollen auf kurzen Wegen zum Ziel kommen.
- Die Anwendung muss gleichermaßen über Maus und Tastatur bedienbar sein.
- Der Benutzer sollte, sofern es sinnvoll ist, sein eigenes Vokabular einbringen können (z. B. bei der Benennung von Objekten). Es sollte möglich sein, individuelle Funktionen (Makros) zu installieren.
- In einem personalisierten Web-Portal kann der Benutzer festlegen, welche Fenster an welcher Bildschirmposition angezeigt werden.
- Ein editierbares Profil ermöglicht es anzugeben, welche News man in einer Mailing-Liste lesen möchte.
- Den Kunden eines Online-Shops wird erspart, persönliche Bestellinformationen bei jedem Besuch eingeben zu müssen. Das System erkennt sie und füllt die entsprechenden Formularfelder selbständig aus.
- Auf einer Website besteht die Möglichkeit, eine HTML- oder Flash-Version anzuwählen.
- Ein Flash-Intro kann übersprungen werden.



Die Symbolleiste eines Programmes kann individuell konfiguriert werden.

4.3 Benutzer-Objekt-Modell

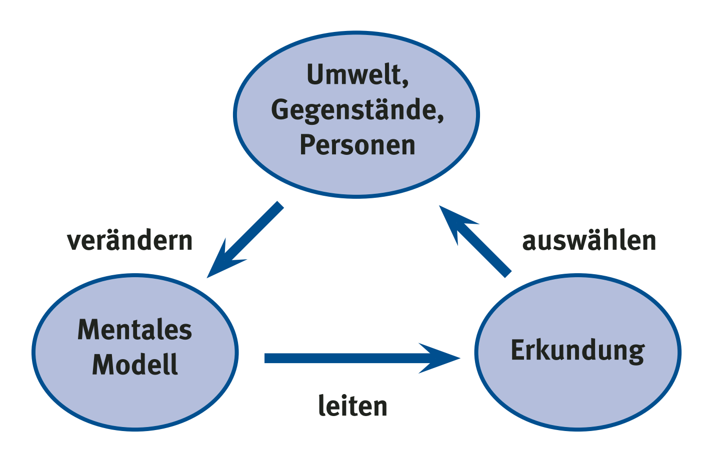
Der Glaube, dass die eigene Sicht der Wirklichkeit, die Wirklichkeit schlechthin bedeutet, ist eine gefährliche Wahnidee. Die sogenannte Wirklichkeit ist das Ergebnis von Kommunikation. [9]

Normalerweise liegt in Software-Entwicklungsprojekten als Ergebnis der Analysephase ein Fachkonzept und ein Datenmodell vor. Jedes dieser Konzepte hat einen anderen Fokus. Das Fachkonzept beinhaltet in der Regel die Beschreibung der Geschäftsvorfälle und der fachlich erforderlichen Funktionen, das Datenmodell beschreibt die Auslegung der Systemfunktionalität, d. h. die Datenstrukturen, Datenoperationen und Datenintegritätsbedingungen.
Da jedoch die Entwicklung von Anwendungssystemen ein komplexes und teures Vorhaben mit vielen Beteiligten und Interessen ist, reichen diese Repräsentationsformen nicht aus.
Neben den beiden oben erwähnten Konzepten muss es ein Benutzer-Objekt-Modell geben. Dieses beschreibt in grafischer Form

- die Objekte, die der Benutzer wahrnimmt,
- die Sichten/Zustände, die auf diese Objekte vorgesehen sind und
- die Interaktionen, die Benutzer mit diesen Objekten haben.

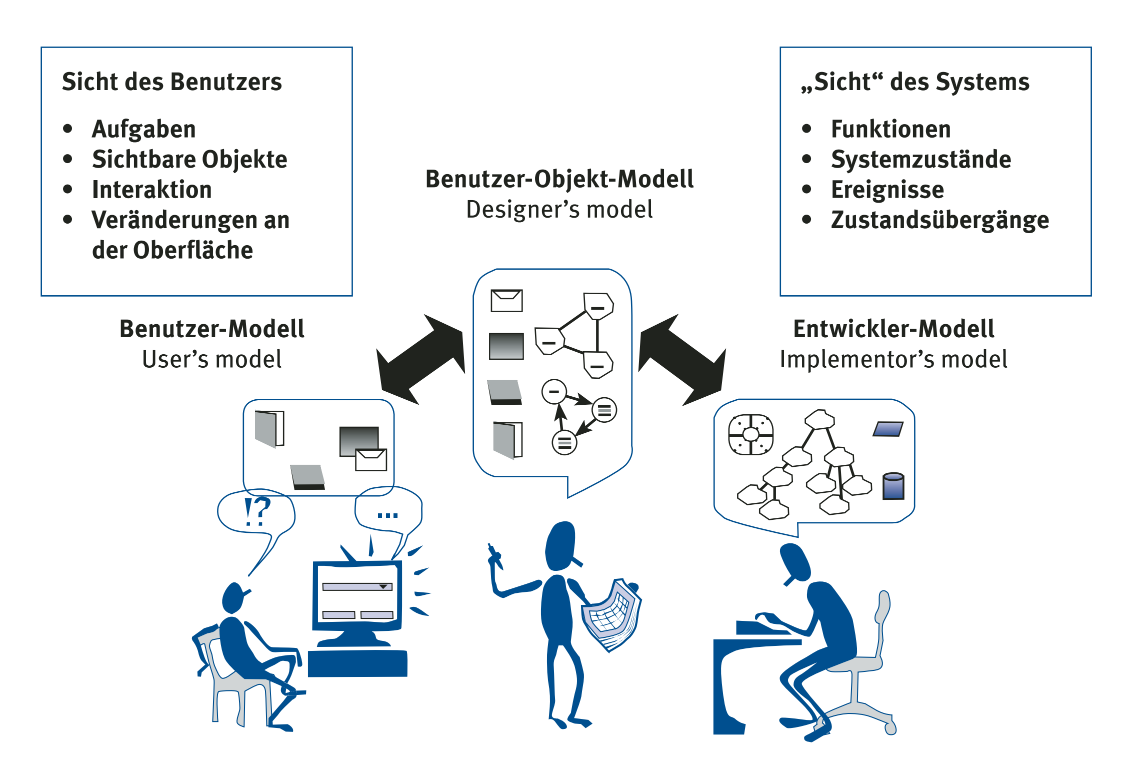
Modellbildung bei zwei Menschen (Tierbesitzerin – Tierärztin) zu ein und demselben Objekt (Katze)

Alle Menschen bilden sich Modelle von der Welt in ihren Köpfen. Diese Modelle beschreiben die Objekte, auf die wir treffen und ihre Beziehungen untereinander. Wann immer Menschen etwas in der Welt begegnet, beziehen sie es unbewusst auf ihr Modell.
Man nennt das Modell, welches der Benutzer einer Software im Kopf hat, das Benutzer-Objekt-Modell. Mögliche Objekte dieses Modells für Büroanwendungen sind Akten, Anträge und Aktionen, die man mit diesen Objekten veranlasst.
Neben dem Benutzer hat auch der Entwickler sein mentales Modell der Arbeit, welches sich auf die Programmierung bezieht. Mögliche Objekte dieses Modells sind Klassen, Attribute und Funktionen für Objekte.
Ziel einer ergonomischen Gestaltung ist es, die Struktur einer Software am Objekt-Modell des Benutzers auszurichten.



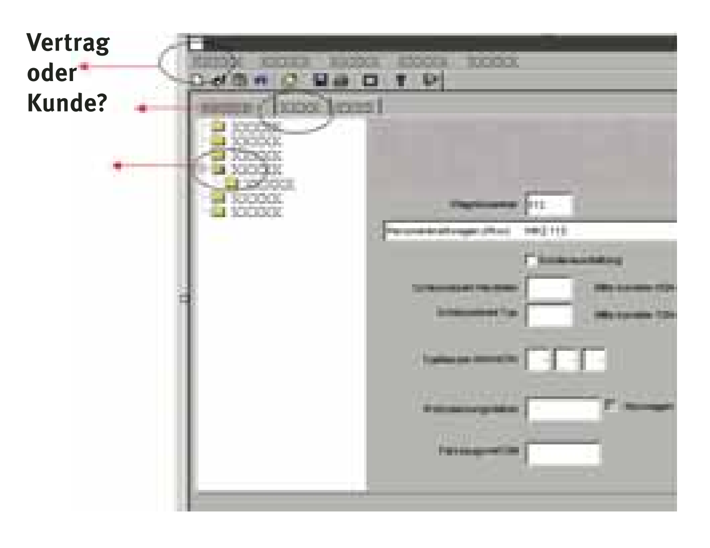
Mentales Modell: Die Vorstellung, die sich der Mensch von Vorgängen in seiner Umwelt/an seinem Arbeitsplatz macht

[9] Paul Watzlawick



Unterschiedliche Funktionsträger haben unterschiedliche Modelle: Alle beteiligten Seiten haben unterschiedliche Intentionen, weshalb es zu Missverständnissen kommen kann.

Die Entwicklung des Benutzer-Objekt-Modells hat das Ziel, die Gegenstände der Handlung aus Benutzersicht zu finden, die Zusammenhänge zwischen den Gegenständen herauszufinden und wie Benutzer mit diesen Gegenständen umgehen. Dabei ist es nicht ungewöhnlich, dass verschiedene Bereiche eines Unternehmens verschiedene Sichten haben (so kann der Innendienst einer Versicherung den Vertrag als zentrales Objekt sehen, wohingegen der Außendienst den Kunden als zentrales Objekt sieht). Das analysierte Modell muss mit allen Benutzergruppen validiert werden.
Dieses Modell ist Basis für grundlegende Entscheidungen der GUI-Entwicklung, d. h. der Gliederung der Anwendung in Ansichten, Menüs und Dialoge. Die Entscheidungen hierfür werden in Projekten meist sehr früh getroffen und haben weitreichende Auswirkungen auf die gesamte Software-Architektur. Aus Sicht der Usability sind diese Aspekte von erheblicher Bedeutung.



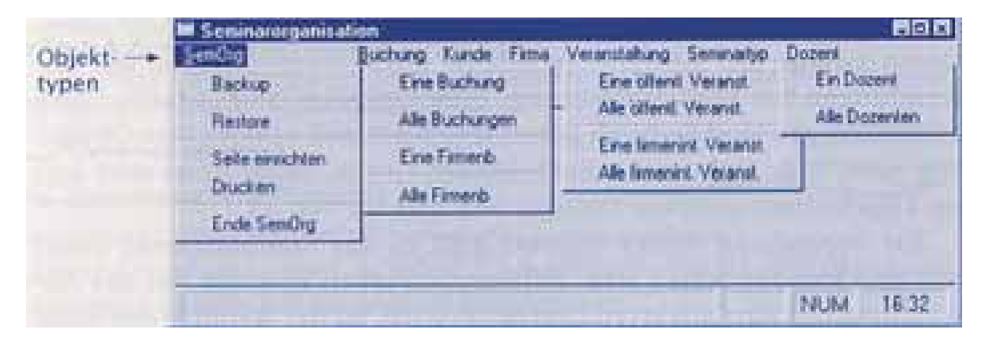
Strukturierung einer Anwendung nach Vertragsarten (z. B. alle Kfz-Versicherungen) oder nach Kunden (alle Verträge eines Kunden) – unterschiedliche Sichtweise eines Innen- oder Außendienstmitarbeiters einer Versicherung.

4.4 Menügestaltung (DIN EN ISO 9241-14)

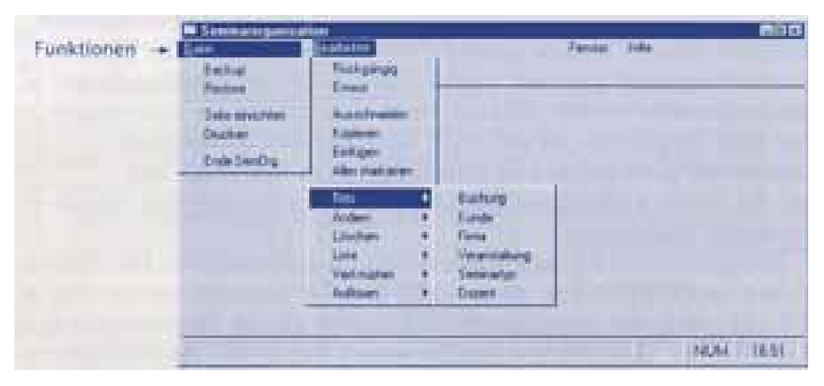
Ein Menü ist die Darstellung von zu Gruppen angeordneten Aktionen und Eigenschaften zur Steuerung des Primärdialogs der Applikation.
Das Menü besteht aus einer überschaubaren Menge von Menüoptionen, aus denen der Benutzer eine oder mehrere auswählen kann.
Die Konzeption des Menüs muss aus dem Benutzer-Objekt-Modell abgeleitet werden. Wenn bekannt ist, mit welchen Objekten die Benutzer welche Aktionen ausführen, ist zu entscheiden, ob man das Menü objekt- oder funktionsorientiert gestaltet oder eine Mischform wählt.

In der Regel denken Menschen zuerst an ein Objekt und dann daran, was sie damit machen wollen.
Bei dem Office-Produkt „Word“ von Microsoft resultiert die Bezeichnung der Hauptmenü-Optionen „Datei“, „Bearbeiten“, „Ansicht“ daher, dass es nur ein Objekt gibt (die Datei), mit dem der Benutzer arbeitet. Dort befinden sich alle Aktionen, die sich auf das ganze Objekt beziehen unter der ersten Menüoption „Datei“, unter der Option „Bearbeiten“ finden sich Aktionen, die Teile des Objektes betreffen usw.

Datei	Bearbeiten	Ansicht ~	Fenster	?
Handlungsoptionen zum **zentralen Objekt** der Anwendung **Beispiel:** • Sichern • Beenden	Handlungsoptionen zu **Bestandteilen** (Unter-Objekten) des zentralen Objekts der Anwendung **Beispiel:** • Kopieren	Handlungsoptionen, die die **Ansicht** des Objektes verändern **Beispiel:** • Sortieren • Teil-Ansicht	Möglichkeiten der Anordnung von **Fenstern** und eine Liste aller Fenster	Beschreibung der Handlungsmöglichkeiten zur Bearbeitung der Objekte und zur Benutzung der Oberfläche



Objektorientiertes Menü
Objekte bilden die Hauptgruppen, Funktionen sind als Druckknöpfe auf den Fenstern oder im Menü angeordnet.

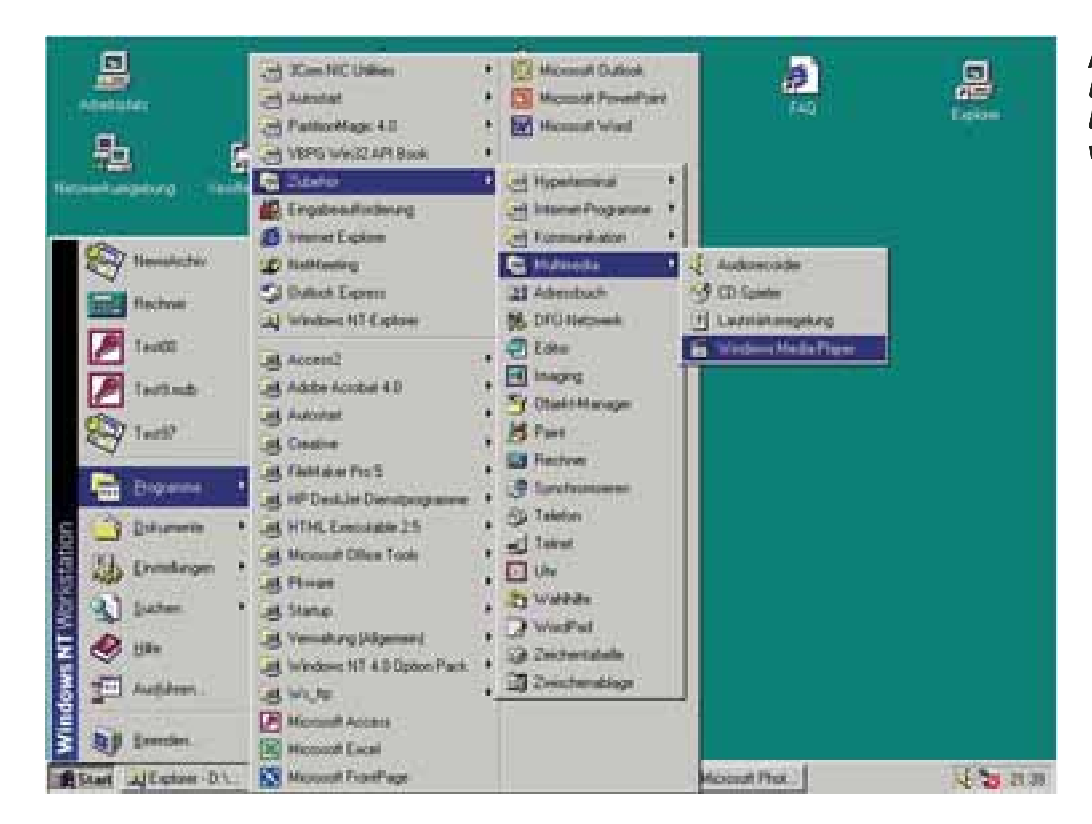


Funktionsorientiertes Menü
Funktionen bilden die Menüpunkte, Objekte werden nach Aktivierung einer Funktion als Unter-Fenster dargestellt.

Quelle: Balzert, H.: Lehrbuch der Softwaretechnik, Spektrum, 1998

Die Zuordnung der Hauptmenüoptionen und den darunter angeordneten Begriffen sollte das mentale Modell der Benutzer von der Anwendung spiegeln.
Welche Lösung für eine Eigenentwicklung die optimale ist, lässt sich mit Prototyping am besten herausfinden. Dieses kann in einem Benutzer-Workshop mit Metaplankarten konzipiert werden.

Es ist zu beachten, dass Benutzer die Auswahl in Menüs nicht nur nach dem Inhalt vornehmen, sondern auch nach ihrem räumlichen Gedächtnis. Deshalb ist auf eine gewisse Symmetrie zu achten, d. h., nicht unter eine Menüoption sechs Einträge und unter einem anderen nur zwei Optionen.



Microsoft Windows NT4: Ungünstige Verschachtelung von Menüs verlangt viel Feinmotorik.

Gestaltungsregeln für Menüs

- Als Menütitel sind maximal 7 +/-2 Oberbegriffe zu wählen, die einen Rückschluss auf die darunter zusammengefassten Funktionalitäten zulassen.

- Die Bezeichnungen sollten möglichst nur aus einem Wort bestehen, möglichst aussagekräftig sein („Sonstiges" nur wenn unvermeidbar).

- Die Reihenfolge von rechts nach links sollte der Hierarchie des Benutzer-Objekt-Modells folgen.

- Für ein einzelnes Menü sind maximal drei Hierarchiestufen zulässig.

- Eine zu tiefe Schachtelung erschwert das Auffinden, erhöht die Zugriffszeit und ist feinmotorisch schwer zu handhaben.

- Je häufiger ein Befehl verwendet wird, desto weiter oben sollte er in der Menühierarchie geordnet sein.

- Bezeichnungen von Menüpunkten, hinter denen sich der Aufruf von Dialogfenstern verbergen, enden mit drei Punkten.

- Bezeichnungen von Menüpunkten, hinter denen ein weiteres Kaskadenmenü folgt, sind mit einem kleinen Dreieck gekennzeichnet.

- Menüeinträge, also Funktionalitäten, die im aktuellen Anwendungskontext nicht verwendet, d. h. nicht aufgerufen werden können, sind inaktiv (disabled) darzustellen.

- Bei mehr als sieben Menüeinträgen werden die Menüeinträge durch Separatoren (optische Trennlinien) getrennt und zu logischen Gruppen zusammengefasst.

- Bei zu vielen (besonders thematisch zusammengehörigen) Optionen sollte geprüft werden, ob eine Verzweigung mit Dialogfenstern oder Registern als Alternative sinnvoll ist. Diese Entscheidungen hängen aber auch vom Benutzer-Objekt-Modell ab.

4.5 Fenster- und Navigationskonzept

4.5.1 Fensterkonzepte

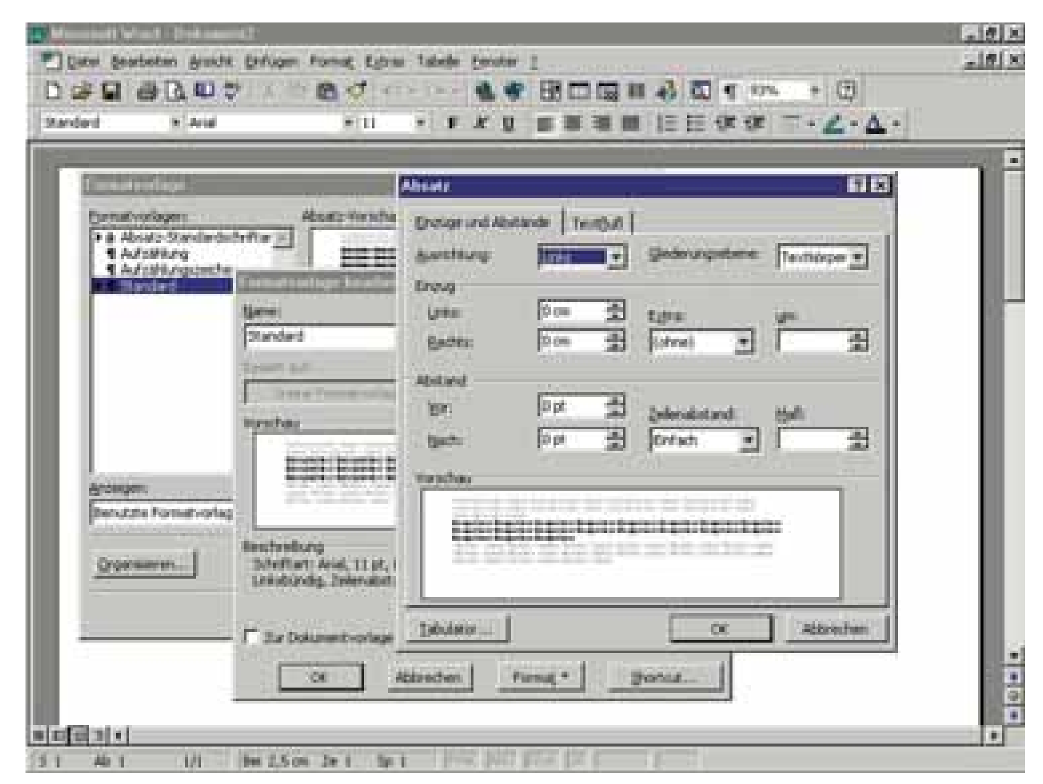
Im Benutzer-Objekt-Modell wird beschrieben, welche Aktionen die Benutzer mit welchen Objekten zur Aufgabenerledigung ausführen. Als weiterer wichtiger Faktor kommt die Sicht auf die Objekte hinzu.

Diese Sichten spielen eine wichtige Rolle bei der Wahl des Fensterkonzepts (SDI/MDI etc.[10]) und für die Hierarchien der Fenster (Hauptfenster, Unterfenster, modale Dialoge).

Im Normalfall sind Fenster unabhängig voneinander zu öffnen und zu schließen. In den meisten Anwendungen enthält jedoch ein Fenster übergeordnete Daten zum Objekt bzw. zu den Objekten, und Nebenfenster liefern Detailinformationen zu dem Objekt und sind deshalb nicht unabhängig.

Die Wahl eines aufgabenangemessenen Fensterkonzeptes ist keine triviale Aufgabe.

[10] SDI / MDI = Single-Document / Multiple-Document Interface



Einsatz mehrerer Fenster zur Bearbeitung einer Aufgabe – die Fenster überlagern einander (das aktuelle „liegt oben auf“ und ist durch die farbliche Hervorhebung der Titelleiste als zur Zeit einzig aktives Fenster zu erkennen; die anderen Fenster sind in der Titelleiste grau gestellt und sind zur Zeit nicht aktiv).

Es gibt hier keine absolut vorgegebenen Konzepte, sondern Richtlinien, aus denen man Mischformen bilden kann, solange sie konsistent und verständlich sind.

Die Auswahl des Fenstermodells sollte sich an den Bedürfnissen, Kenntnissen und v. a. der Computererfahrung der Benutzer ebenso orientieren wie an den Erfordernissen der Anwendung.

Die Entscheidung, welches Fensterkonzept gewählt wird, hat große Auswirkungen auf die Gebrauchstauglichkeit und hängt eng mit dem Benutzer-Objekt-Modell zusammen. Hier kann man die richtige Entscheidung nur mit Methoden des Usability Engineering treffen, wie z. B. mittels eines moderierten Benutzer-Entwickler-Workshop.

Modale Dialoge
Ein weiterer Aspekt, der mit der Fenstertechnik zusammenhängt, sind modale Dialoge.
Modale Dialoge unterscheiden sich sowohl bzgl. des Layouts, des Inhalts, des Verhaltens als auch des Verwendungszwecks von anderen Fenstern.
Modale Dialoge werden immer von Fenstern aus aufgerufen (Ausnahme: Meldungsfenster). Die Aktionen, die dem Benutzer innerhalb des modalen Fensters angeboten werden, stehen in direkter Abhängigkeit zum aufrufenden Fenster sowie ggf. sogar zum aktuellen Anwendungskontext, also der dort durchgeführten Aktionen und selektierten Daten. Modale Dialoge sind erforderlich, wenn für die Anwendung eine bestimmte Information oder Bearbeitung unbedingt benötigt wird, bevor die Bearbeitung fortgesetzt werden kann.

Aus Sicht der Software-Ergonomie sind modale Dialoge nur einzusetzen, wenn es unbedingt nötig ist, z. B. wenn die Arbeit im aktuellen Zustand verboten ist (Passwort, Parameter fehlen) oder Information zur Kenntnis genommen werden soll (Info, Meldung, Warnung). Darüber hinaus kann ein modaler Dialog zur Abfrage notwendiger Einstellungen (Abfrage, Druckdialog) benutzt werden. Zu empfehlen ist, dass die Aktion, die vom Benutzer gefordert wird, möglichst positiv zu bestätigen ist.

Bei Web-Applikationen gibt es die oben beschriebene abhängige Fensterhierarchie nicht. Die Struktur findet sich im Site-Konzept. Hier gibt es Dokumente und Verknüpfungen die eine Art von Hypertext[11] bilden. Der Raum, in dem man sich bewegt, ähnelt nicht wie bei den Fensterkonzepten der GUI-Anwendungen einem Buch, in dem man linear blättert, sondern mehr einem dreidimensionalen Raum. Kleine Informationseinheiten bilden eine Informationsarchitektur (Wurzel, Baum, Netz…). Die Struktur besteht aus Knoten (Informationseinheit) und Kanten (Hyper-Links), der Benutzer kann frei durch diesen Raum navigieren.

4.5.2 Navigation/Site Design

Die Navigation beschreibt die Bewegung des Benutzers durch die Sichten des Systems. Dies ist aus Sicht der Software-Ergonomie ein zentraler Aspekt, der eng mit dem Benutzer-Objekt-Modell zusammenhängt. Hierzu ist auf jeden Fall auch zu ermitteln, ob es verschiedene Benutzergruppen gibt, die unterschiedliche Sichten auf die Anwendung haben. Für jede diese Benutzergruppen muss eine gemeinsam verständliche Navigation gefunden werden. Dazu sind während der Analyse die dynamischen Abläufe der Dialogführung zu strukturieren und mit Benutzern abzustimmen.

Um eine angemessene Navigation zu erreichen, muss man jedoch konzeptionell den zu unterstützenden Arbeitsablauf ermitteln.

Zur Strukturierung konventioneller GUI-Software ist es sinnvoll, zwischen zwei Ebenen zu unterscheiden:

- Allgemeiner Dialog = Navigationsdialog zwischen den einzelnen Sichten (hier Fenster oder Masken) bis zur eigentlichen Aufgabenbearbeitung.

- Anwendungsspezifischer Dialog = Bearbeitungsdialog zur Eingabe oder Manipulation von Daten auf der jeweiligen Sicht (Sprünge zwischen Datenfeldern, auf einer Maske oder zu Steuerelementen).

Für eine Intranet-Gestaltung spricht man von

- globaler und

- lokaler Navigation.

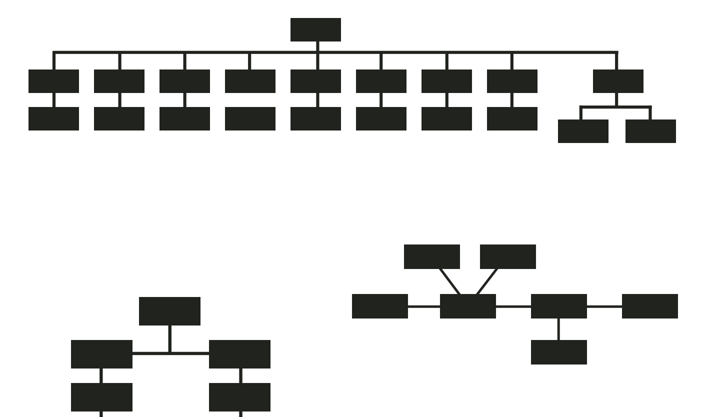
Globale Navigationssysteme bilden die oberste Ebene der Navigation ab. Sie stellen Links zur Verfügung, die für die gesamte Web-Site gelten und werden i. d. R. über ein grafisches Navigationsmenü dargestellt.

Das lokale Navigationssystem bietet Links für Gruppen von Web-Pages, die inhaltlich einem bestimmten Bereich angehören, z. B. die Unterseiten für den Bereich „Angebote“. Wichtig ist, dass das globale Navigationssystem erhalten bleiben muss.

[11] Hypertext: Dokumentformat, welches ein Verknüpfen von Informationsbausteinen und Querverweisen (Hyperlinks) ermöglicht. Die Verbindungen zwischen den einzelnen Knoten erlauben das (assoziative) Navigieren in einem Hyper-Dokument.

Zusätzlich muss die sogenannte Ad-Hoc-Navigation vorgesehen werden. Diese Navigation ist abhängig vom aktuellen Inhalt der Seiten, d.h., es werden Links angeboten, die nur für eine spezielle Seite gelten. Dies können z. B. in den Inhalt eingebettete Links sein. Besser sind allerdings externe Links, die z. B. unterhalb eines Textes angeordnet werden. Studien haben ergeben, dass eingebettete Links den Benutzer bei der Suche nach Informationen eher verwirren.

Einschränkungen der Navigation/Interaktion bei GUI-Applikationen werden durch unterschiedliche Dialogmodi realisiert. Solche Einschränkungen können fachlich erforderlich sein, sollten dem Benutzer aber deutlich visualisiert werden. Modi sind z. B. der Änderungsmodus, der Einfügemodus usw. Der Navigation in der Web-Software liegt das Site-Design zugrunde. Dieses kann verschiedenartig aussehen:



Unterschiedliche Site-Strukturen

Je nach Art der Informationen kann eine bestimmte Seitenstruktur optimal sein. Die meisten Web-Seiten sind jedoch organisch gewachsen und nicht systematisch geplant. Dies führt dazu, dass auch die Navigation nicht immer nachzuvollziehen ist. Für ein Intranet gilt:

- Klare Strukturen für eine einfache Orientierung (Erleichterung der Bildung eines mentalen Modells).
- Die Breite der Struktur (Optionen auf einer Ebene) sollte nicht mehr als 10 Optionen umfassen (Gedächtnisspanne 7 +/- 2 Einheiten).
- Die Tiefe der Struktur sollte nicht mehr als drei bis vier Ebenen umfassen.

Für die Site-Gestaltung sollte man folgende Aspekte beachten:

- Jede Seite kann innerhalb von drei Mausklicks angesteuert werden (Faustregel).
- Navigations- und Informationsseiten trennen.
- Festlegen, welche Art von Information auf welcher Ebene zu finden ist:

  1. Ebene: Wo bin ich, wohin kann ich gehen, wie komme ich dahin?

  2. Ebene: Zweckorientierte Fragen: Was ist, wenn ich nicht sicher bin, was ich suche?

  3. Ebene: Produkt- oder zielgruppenorientierte Fragen: Wie kann ich etwas finden?

- Anzeige des Kontextes auf jeder Seite.
- Organisations-/Firmenname auf allen Seiten anzeigen als Titel oder Überschrift
- Anzeige des Navigationspunkts (Pfad), über den der Benutzer auf diese Seite gelangt ist bzw. Anzeige des Standortes innerhalb der Struktur.
- Einheitlichkeit (Konsistenz) im Aufbau des gesamten Web-Auftritts.
- Möglichkeit des schnellen Zurücknavigierens.
- Anzeigemöglichkeit auch größerer Mengen von Optionen bzw. Links.
- Verständliche, aussagekräftige Links.
- Die Struktur der Links auf der Homepage möglichst nicht verändern, sondern neue Einträge auf der zweiten Ebene hinzufügen (mentales Modell wird vor allem über die Homepage gebildet).

In der Norm DIN EN ISO 14915-2 (Software-Ergonomie für Multimedia – Benutzungsschnittstellen) heißt es zur Navigation in Baumstrukturen:

„Falls für die Aufgabe geeignet, sollte das System dem Benutzer Mittel bieten, um:

- sich auf einer Ebene der Struktur vor und zurück zu bewegen;
- in der Struktur nach oben und unten zu gehen;
- sich in größeren Schritten zu bewegen (z. B. von einer Seite zur nächsten oder zu anderen Gruppen von Inhaltsblöcken);
- sich zur ersten Schicht/Hauptschicht zu bewegen;
- zum Anfang oder zum Ausgang der Struktur zu gehen und
- sich zu einem Inhaltsverzeichnis oder Index zu begeben."

Des weiteren finden sich in der DIN EN ISO 14915-2 folgende Empfehlungen:

Abschnitt der DIN EN ISO 14915-2	**Gestaltungsempfehlungen**
Navigationsstrukturen linear, Baum, Netz	Hier finden sich Aussagen zur Gliederung von Inhalten, wie man Benutzer beim Navigieren unterstützten kann, zur Verwendung von Navigationsmetaphern, über lineare, Baum- und Netzwerkstrukturen
Links	Hier finden sich Regeln für die allgemeine Benutzung und Darstellung und das Verhalten von Links
Navigationsfunktionen	Hier finden sich Empfehlungen zur Benutzerunterstützung allgemein und zu Navigationsmethoden zwischen und innerhalb von Präsentationssegmenten (Vor- und Zurück-Funktion etc.)

4.6 Benutzerführung (DIN EN ISO 9241-13)

Unter Benutzerführung versteht man zusätzliche Informationen, die über den regulären Benutzer-Computer-Dialog hinausgehen und die entweder auf Verlangen des Benutzers oder automatisch durch das System präsentiert werden. Diese Informationen zur Benutzerführung sollten sich klar von anderen Ausgabeinformationen unterscheiden.

4.6.1 Rückmeldungen

- Jede Eingabe des Benutzers sollte eine wahrnehmbare Rückmeldung vom System erzeugen.
- Rückmeldungen sollten unaufdringlich sein und den Benutzer nicht von seiner Arbeitsaufgabe ablenken.
- Das System sollte seinen Zustand eindeutig anzeigen, sobald sich der Zustand oder Modus ändert.
- Wenn ein Benutzer ein angezeigtes Objekt auswählt, um eine Aktion an oder mit dem Objekt auszuführen, sollte das Objekt hervorgehoben werden.
- Die erfolgreiche Ausführung von Anweisungen der Benutzer sollte zurückgemeldet werden.
- Wenn die Ausführung einer Benutzeranweisung nicht sofort erfolgt, dann sollte das Dialogsystem mitteilen, dass die Anweisung angenommen ist. Außerdem sollte das Dialogsystem auch die erfolgreiche Beendigung der Anweisung anzeigen.
- Durch das System initiierte Nachrichten der Benutzerführung müssen entfernt werden, wenn der Zustand nicht mehr zutrifft.

4.6.2 Statusanzeigen

- Jeder Typ von Statusinformation sollte an gleichbleibender Stelle auf dem Bildschirm (im Fenster) angeordnet werden.
- Wenn das Dialogsystem die Eingabe durch den Benutzer gesperrt hat, sollte dem Benutzer ein Hinweis gegeben werden, der diesen Zustand anzeigt.
- Wenn ein System oder eine Anwendung Modi hat, sollte der Benutzer den augenblicklichen Modus von den anderen Modi unterscheiden können.

4.6.3 Fehlermanagement

Menschen machen Fehler und aus Fehlern lernt man. Dies bedeutet für Software, dass der Benutzer in der Lage sein sollte, die letzte Operation rückgängig zu machen, sofern es die Aufgabe erlaubt. Bei der Arbeit am Bildschirm kann es aus verschiedenen Gründen zu Fehlersituationen kommen:

- Keine Zugriffsberechtigung des Benutzers.
- Fachlich falsche Eingabe oder
- unsinnige Eingabe.
- Wenn Benutzeraktionen kritische Auswirkungen haben können und sich nicht rückgängig machen lassen, sollte eine Warnung oder eine Aufforderung zur Bestätigung vorgesehen werden, um den Benutzer auf die Konsequenzen aufmerksam zu machen, ehe die verlangte Aktion ausgeführt wird.
- Wenn Systemstörungen vorhersehbar sind, sollte auf das mögliche Problem hingewiesen werden, bevor die Störung eintritt.
- Wenn der Benutzer ein Programm beenden oder das System verlassen will, sollte das System den Status von Dateien und von ausstehenden Transaktionen überprüfen. Wenn Benutzerdaten verloren gehen können oder ausstehende Transaktionen eventuell nicht abgeschlossen werden können, sollte eine Nachricht angezeigt werden, die vom Benutzer eine Bestätigung verlangen.
- Plausibilitätskontrollen und Fehlermeldungen sollten bei der Eingabe so früh wie möglich erfolgen, d. h., auf Zeichen- oder Feldebene.
- Der Benutzer soll Eingaben modifizieren oder löschen können, ehe eine Aktion ausgeführt wird. Wenn dies ohne Systembeeinträchtigung oder Datenverlust möglich ist, sollte der Benutzer die Möglichkeit haben, in der Ausführung befindliche Operationen zu unterbrechen.
- Wenn eine Fehlerkorrektur vom System vorgesehen ist, sollte eine Sicherheitsabfrage oder Warnmeldung vorgesehen werden, die dem Benutzer die geplante Korrektur anzeigt.

Fehlerbehandlung durch den Benutzer

- Wenn die Arbeitsaufgabe eine Fehlerbehandlung durch den Benutzer verlangt, sollte der Dialog Mittel (Informationen und/oder Funktionen) bereitstellen, die es dem Benutzer erlauben, den Dialog fortzusetzen.
- Wenn die Fehlerkorrektur dem Benutzer überlassen wird, sollten Hilfsmittel für die Fehlerkorrektur vorgesehen werden.

Fehlermeldungen

Bei Anzeige einer Fehlermeldung ist eine außergewöhnliche Situation eingetreten, die bewirkt, dass die Aktion im rufenden Dialog nicht fortführbar ist. Die Fehlerursache muss zuvor beseitigt werden.

- Fehlermeldungen sollten vermitteln, was falsch ist, was die Ursache des Fehlers ist und welche Korrekturmaßnahmen ergriffen werden können.
- Das System sollte es dem Benutzer erlauben, Fenster mit den Fehlermeldungen zu verschieben, wenn sie wahrscheinlich relevante Aufgabeninformation verdecken.
- Die Fehlermeldung beinhaltet immer einen Detail-Button, über den ein Anwender zusätzliche Informationen erhält, die die Hotline ggf. für die Lösung des Problems benötigt.

In der Praxis ist es sinnvoll, ein Konzept für diesen Bereich zu erstellen. Dort wird zwischen den auftretenden Fehlern unterschieden und die jeweilige Art und Form der Rückmeldung dokumentiert. Dies können sein:

- Systemfehler, z. B. Verbindung zum Server existiert nicht mehr
- Datenbankfehler, z. B. die Datenbank ist nicht verfügbar
- Applikationsfehler, z. B. falsche oder fehlende Eingaben

Der Benutzer ist lediglich in der Lage, Fehlerursachen zu beheben, die auf falsche Aktionen seinerseits, bzw. auf falsche oder fehlende Eingaben zurückzuführen sind. System- oder Datenbankfehler kann der Benutzer im Allgemeinen nicht beheben.

Mögliche Regelungen:

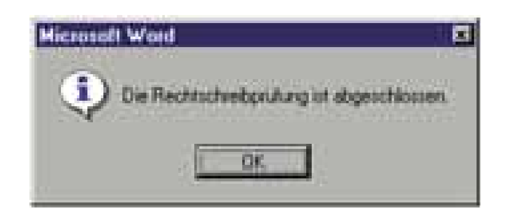
- System- und Datenbankfehler

Nach Bestätigung der Fehlermeldung wird in den rufenden Dialog zurückgekehrt. Der Anwender kann keine Aktionen und Eingaben mehr durchführen, sondern kann den Dialog bzw. die Applikation lediglich schließen und die durchgeführten Modifikationen abbrechen.

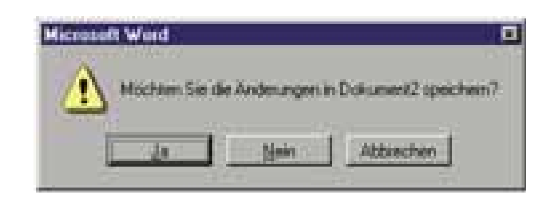
- Applikationsfehler

Handelt es sich um einen Applikationsfehler, so wird für die Korrektur in den Dialog zurückverzweigt, der den Fehler verursacht hat.
Handelt es sich um fehlende oder falsche Dateneingabe oder -auswahl, so ist der Cursor auf dem Dialogelement zu positionieren, das die Fehlermeldung ausgelöst hat. Sind mehrere Dialogelemente betroffen, so ist der Cursor auf dem Element zu positionieren, das gemäß der Tab-Order an vorderster Stelle steht.

Nachrichtenfenster mit informativem Charakter



Nachrichtenfenster mit warnendem Charakter



Nachrichtenfenster mit Nachricht über nicht fortführbaren Dialog in einer außergewöhnlichen Situation



Klassifizierung von Rückmeldungen

4.7 Gestaltungsempfehlungen für die Eingabe in Formulare (DIN EN ISO 9241-17)

In vielen Verwaltungsbereichen wird mit Formularen gearbeitet, weshalb auch zahlreiche Anwendungsprogramme diese Metapher übernommen haben.
Die Benutzer setzen dabei in gekennzeichnete Felder Daten ein und modifizieren diese. Dies ist eine besondere Form der Dialoggestaltung.
In der DIN EN ISO 9241–17 wird dies so definiert:
Dialoge mittels Bildschirmformularen sind Dialoge, in welchen der Benutzer in Felder in einem „Formular", das durch ein System dargestellt wird, einsetzt oder modifiziert, und bei diesem Tun wird er über das System informiert, dass der Eintrag eingegangen und akzeptiert worden ist (üblicherweise am Ende eines Absatzes oder am Ende das gesamten Formulars). Das System kreiert oder aktualisiert dann den Datenbestand, der mit dem Formular assoziiert wird.

Zu dieser Form der Dialoggestaltung finden sich zahlreiche Gestaltungshinweise:

- Eingabe-Dialoge sollten so entwickelt sein, dass der Benutzer Information oder Daten in einer Form eingeben kann, die für ihn „natürlich" ist, und ohne dass er sich Gedanken darüber machen muss, wie der Computer diese Information oder Daten verarbeitet.

- Der Benutzer sollte zu jeder Zeit die Kontrolle über den Dialog behalten, sollte schnell Fehler erkennen können und sollte nicht mehr Information als nötig eingeben müssen, um seine Arbeit erfolgreich auszuführen.

- Der Benutzer sollte die Möglichkeit haben, das Formular zu löschen und ganz neu zu beginnen, sollte einzelne Einträge löschen können und jeden Eintrag überprüfen und ändern können, bevor der abschließende Eintragungsvorgang ausgeführt wird.

- Es sollte dem Benutzer Feedback gegeben werden, damit er Informationen darüber erhält, wie er den Dialog kontrollieren kann, damit er Fehler erkennen und die nächste Handlung festlegen kann.

- Es sollten für den Benutzer, der Zugang zu verschiedenen Bereichen des Formulars benötigt, Navigationsmethoden vorgesehen sein. Diese sollten den verschiedenen Benutzertypen und deren Arbeitsstil angepasst sein.

- Verwendung von klaren Maskentiteln.

- Anweisungen für gelegentliche Benutzer, wie das Formular komplettiert, gesichert und überschrieben werden kann.

- Übereinstimmung des schriftlichen Formulars mit der Maske bzw. umgekehrt sowie Gruppierung nach Wichtigkeit, Funktion usw.

- Zwingende Felder sollten zuerst platziert werden.

- Optionale und zwingende Felder unterscheidbar gestalten, ebenso wie Nur-Lesefelder.

- Anordnung der Eingabefelder so, dass Tastenbetätigungen minimiert werden.

- Verwendung aussagekräftiger Feldbezeichner.

- Mehrseitige Formulare nach der „1 von 3"-Methode kennzeichnen.

- Das Navigieren im Formular soll mit allen Eingabemedien zu allen Feldern möglich sein.

- Keine interdependenten Feldbeziehungen.

- Bei Auswahl aus einer Optionen-Liste ist die optische Unterscheidung der gewählten Angabe erforderlich sowie zügige Navigationsmöglichkeit bei langen Listen bzw. bei 2–5 Optionen die Auswahl mittels Tasten.

- Wenn eine Fehlerkorrektur verlangt wird, darf ein Benutzer nicht zum nächsten Formular kommen können.

- Der Formularaufruf sollte über das Menü und die Benennung eines Namens erfolgen können.





Bildschirmformulare und Papierformulare sollten ähnlich aufgebaut sein.

4.8 Gestaltung von Intranets

„If you can use a site, then you will return. If you can’t, you just leave (with one click).” [12]

Ein Intranet kann man als unternehmensinternen Online-Dienst bezeichnen, der auf Internet-Technologien basiert und durch ein plattformunabhängiges Navigationssystem (Browser) benutzungsfreundlich bedient werden kann. Dieses System dient in der Regel der Verbesserung der internen Kommunikation, dem Informationsaustausch und der schnelleren Abwicklung interner Verwaltungsprozesse. Darüber hinaus haben sich Intranets als Begleitmaßnahme zur Förderung der Mitarbeitermotivation bewährt, denn durch die geschaffene Transparenz kann Vertrauen entstehen, was sich positiv auf das Arbeitsklima auswirkt.
Generell ist festzustellen, dass bei erfolgreichen Intranetprojekten nicht nur in eine gute Software- und Netzinfrastruktur investiert wird, sondern immer häufiger in die Integration der Mitarbeiter in die Gestaltung des Intranets.

Es gibt zwei wesentliche Unterschiede zwischen Internet- und Intranet-Seiten:
Bei externen Internetseiten ist ein wesentliches Ziel, die Kontaktzeit mit dem Benutzer möglichst lang zu halten. Dieses ist bei Intranets nicht das Ziel. Hier geht es darum, den Mitarbeitern eine große Menge an Informationen so zur Verfügung zu stellen, dass diese effektiv und effizient gefunden werden. Intranet-Benutzer werden im Gegensatz zu externen Internet-Site-Benutzern schnell zu Experten, da sie im optimalen Fall das Intranet jeden Tag nutzen. Dies bedeutet, dass je mehr Aufgaben online erledigt werden, die Produktivität steigt.

[12] Jacob Nielsen

Ein weiterer Unterschied zu einer Internet-Site ist, dass die Benutzer im Intranet die Mitarbeiter sind. Im Gegensatz zu externen Kunden einer Web-Site haben diese Vorwissen über das Unternehmen und bestimmte Erwartungen.
Ist ein Aufbau der Site anhand der Organisationsstruktur des Unternehmens für eine Internetseite generell nicht zu empfehlen, ist dies für Intranet-Applikationen positiv zu bewerten. So kann es sinnvoll sein, die Grobgliederung nach Abteilungsbezeichnungen vorzunehmen.

Viele Gestaltungsregeln aus dem konventionellen GUI-Bereich, die in den Kapiteln 3 und 4 beschrieben sind, lassen sich z. B. auf die Intranet-Gestaltung übertragen. Man sollte sich, bei der Verwendung von GUI-Elementen für Web-Software an die gebräuchliche Verwendung dieser Elemente halten. So sollten z. B. Register (Karteikarten) nicht als funktionsauslösende Buttons programmiert sein, da dieses für Benutzer nicht erwartungskonform ist. Für neue browserspezifische Gestaltungselemente, wie z. B. Vor- und Zurück-Buttons müssen entsprechende Regeln aufgestellt und in einem Styleguide dokumentiert werden.
Das Fehlen einer einheitlichen Navigation ist eines der großen Nutzungsprobleme in großen Intranets. Wenn Bereiche eines Intranets von unterschiedlich zuständigen Abteilungen unterschiedlich gestaltet sind, erfordert dies eine ständige Neuorientierung der Benutzer.

Verschiedene Untersuchungen haben ergeben, dass

- beim Online-Shopping zwischen 20 % und 80 % der Einkäufe einfach deshalb schiefgehen, weil die Kunden mit dem schlechten Design der Shop-Seiten nicht zurecht kommen,
- 27 % aller Webaktivitäten beim Bezahlvorgang abgebrochen werden,
- jeder zweite Online-Nutzer durch blinkende Flash-Intros genervt ist,
- sich Surfer im Durchschnitt nur neun Sekunden Zeit lassen, um den gewünschten Inhalt zu finden.

4.8.1 Inhalts- und Navigationsmodell

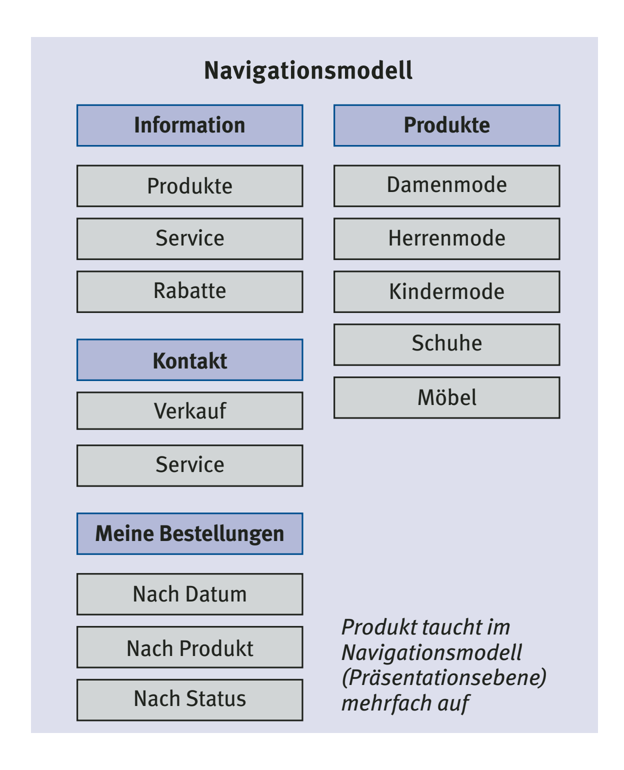
Grundsätzlich müssen bei der Intranet-Gestaltung zwei Schritte bedacht werden:

- Konzeption der Inhalte
- Navigationsmöglichkeiten (durch den Inhalt)

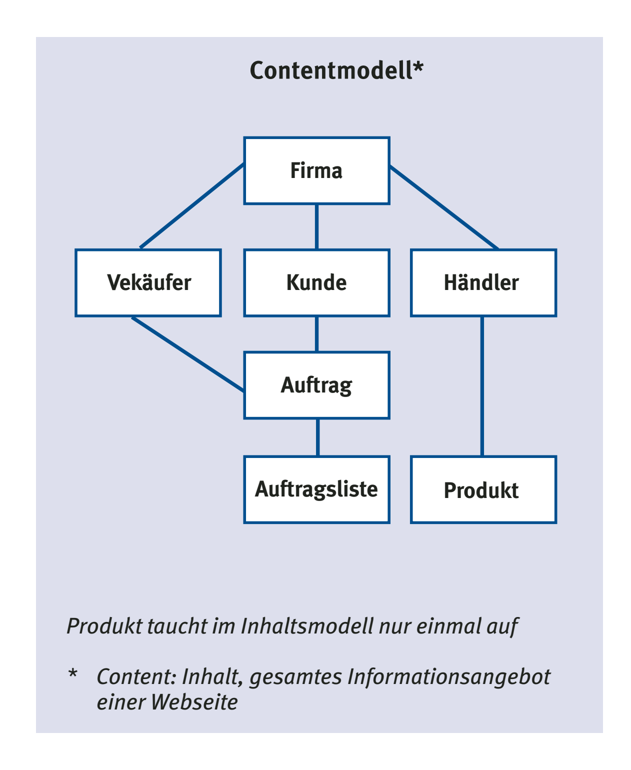
Nur eine gute Struktur des Informationsangebots im Intranet ermöglicht effizientes Arbeiten. Produktivitätssteigerungen lassen sich durch Übersichtlichkeit und Orientierungsmöglichkeiten gewährleisten, d. h., nur wenn die Informationsbeschaffung über das Intranet schneller funktioniert als über herkömmliche Wege, wird dieses Medium genutzt werden. Darüber hinaus sollte es für die Nutzer einen Mehrwert geben.

Eine Struktur, die ein schnelles Auffinden von Informationen ermöglicht, erreicht man durch die Berücksichtigung folgender Aspekte:

- Gewohnte Strukturen nutzen; z. B. Dokumente für Vorlagen (PP oder Word) finden sich im Bereich, der so heißt wie die Abteilung, in der es Formulare gibt.
- Informationen, die nicht direkt einer Kategorie zugeordnet werden können, sollten losgelöst von der Struktur zu finden sein.
- Informationen, die sehr oft benötigt werden, sollten weit oben in der Struktur zu finden sein.



Flexible Präsentationsebene



Relativ stabile Präsentationsebene

Steht die Struktur (Content/Inhalt), geht es im nächsten Schritt darum, die Navigation durch diese Inhalte zu konzipieren. Für das Navigationskonzept einer Intranet-Site sollten folgende Faustregeln beachtet werden:

- Allgemeine Informationen und detaillierte Informationen sind auf unterschiedlich tiefen Ebenen zu platzieren. Dabei ist z. B. darauf zu achten, dass Dokumente, z. B. zum Ausdrucken oder Downloaden, ab der 3. Ebene schwer auffindbar sind.
- Jede Seite ist innerhalb von maximal drei Mausklicks ansteuerbar.
- Von der Homepage sollten nur die Seiten direkt unterhalb der Homepage in der Navigationsleiste angezeigt werden.
- Von den Seiten der 1. Ebene muss es möglich sein, die Homepage, die anderen Seiten der 1. Ebene und die untergeordneten Seiten aufzurufen.
- Von den untergeordneten Seiten muss es möglich sein, die benachbarten Seiten, die übergeordneten Seiten und die Homepage anzusteuern.
- Für eine schnelle Nutzung der Navigation ist besonderer Wert auf die Gestaltung der Navigationshilfen wie Sitemap, Index, Inhaltsübersicht zu legen.

Neben den für die Benutzer interessanten Inhalten und der intuitiven Navigation ist die Aktualität der Daten ein Grund für die Nutzung des Intranets. Dazu gehört:

- Sicherstellen, dass keine alten Informationen im Intranet vorhanden sind, durch Einrichtung, die alte Inhalte nach Verfallsdatum in das Archiv verschiebt.
- Mitarbeiter können in regelmäßigen Abschnitten zum Intranet Feedback geben. Änderungswünsche sollten zeitnah umgesetzt werden.
- Hinweis darauf, was es Neues im Intranet gibt.
- Ein Newsticker, der über Neuigkeiten im Unternehmen informiert.

Des weiteren kann folgendes angeboten werden:

- Diskussionsforen
- Angebote in Fremdsprachen für ausländische Mitarbeiter
- Lageplan/Anfahrtsskizze/Raumplan des Unternehmens
- Organigramm des Unternehmens
- Buchen von Konferenzräumen
- Auf jeder Seite sollte die E-Mail-Adresse des Webmasters/ Intranetverantwortlichen angegeben sein mit Datum der letzten Aktualisierung
- Benutzer sollen mit den Autoren von Texten direkt und sofort Kontakt aufnehmen können
- Mitarbeiter-Umfragen
- Zugriff auf Texte aus dem Archiv
- Interne Mailing-Listen zu speziellen Themen
- Newsletter-Bestellung zu speziellen Bereichen
- Möglichkeit, Vorlagen, Texte, Formulare, Listen usw. herunterladen
- Über das Intranet Zugriff auf das www
- Möglichkeit der Anmeldung zu Seminaren, Veranstaltungen
- Nur als Option für geübte Benutzer: persönlich konfigurierbarer Bereich

Die Nielsen/Norman Group[13] stellte bei der Präsentation der 10 besten Intranets der USA im Jahr 2001 fest, dass gute, d. h., gut gestaltete und gut zu benutzende Intranets unabhängig von Branche, Betriebsgröße und Betriebsart sind und dass auch mit schmalem Budget gute Ergebnisse zu erzielen sind. Demnach geht der Trend zu

- Einfachheit und
- Standardisierung (der Navigation über verschiedene Einheiten!)[14]

4.8.2 Die Suchfunktion

Die Nutzungsqualität eines Intranets ist wesentlich davon bestimmt, dass die Benutzer schnell und einfach bestimmte Informationen finden, die sie suchen.

Man bietet dabei abhängig von der Art der Suche zwei verschiedene technische Lösungen an.

Use Szenario	Technische Lösung
Gerichtete Suche (z. B. einen Urlaubsantrag) Ungerichtete Suche (wer macht was im Unternehmen)	Suchfunktion (einfaches Eingabefeld und/oder mehrere Eingabefelder zur Einschränkung der Suche) Browsing (Vorwärts- und Rückwärtsblättern über die Browse Buttons)

Allerdings ist vor der Konzeption einer benutzergerechten Suchfunktion eine Prüfung der Inhalte und deren Gliederung erforderlich. Denn in einer inkonsistent organisierten Struktur findet man einzelne Inhalte schwerer.

Empfehlungen zur Gestaltung der Suchfunktion

- Grundsätzlich gilt es, die Suche so einfach wie möglich zu gestalten und eine „Fortgeschrittenen-Suche“ (nicht auf der ersten Seite) über einen Link verfügbar zu machen.
- Eine eingeschränkte Suche führt meist zu mehr Verwirrung als Unterstützung.
- Suche setzt Standardisierung von Inhaltsbeschreibungen voraus.
- Suchfunktionalität und Inhaltskategorien müssen sich ergänzen.
- Dies bedeutet, wenn der Inhalt chaotisch organisiert ist, wird auch die beste Suche dem Benutzer kein besseres Verständnis sichern.
- Kurze Trefferlisten.
- Treffertrail (Pfadanzeige).

[13] „Die“ Usability Experten aus USA, http://www.useit.com

[14] J. Nielsen: 10 Best Intranet Designs of 2001 (www.useit.com/alertbox)

4.8.3 Zehn Faustregeln für die Website-Gestaltung

Eine sehr einfache und schnelle Methode, um Websites zu überprüfen bzw. benutzungsgerecht zu gestalten, ist die Anwendung von Heuristiken. Heuristiken sind von Experten zusammengestellte Faustregeln für Websites. Allerdings enthalten sie keine Regeln für die spezielle Anwendungssituation, sollten also nicht ohne eine zusätzliche Benutzerbefragung angewandt werden.
Im Folgenden die von Jacob Nielsen 1994 entworfenen Kriterien:

1. Sichtbarkeit des Systemzustandes
Das System sollte zu jedem Zeitpunkt den Benutzer darüber informieren, womit es sich gerade beschäftigt. Das System sollte mit einer angemessenen Antwort (d. h. verständlich) in vernünftiger Zeit (d. h. innerhalb weniger Sekunden) reagieren.

2. Übereinstimmung zwischen System und der Realwelt
Das System sollte die Sprache des Anwenders sprechen, mit Worten, Phrasen und Konzepten, welche dem Benutzer vertraut sind. Es sollten weniger system-orientierte Meldungen, insbesondere keine Codes verwendet werden. Zusätzlich sollte den Konventionen der realen Welt Rechnung getragen werden. Die Informationen bzw. Meldungen müssen in natürlicher und logischer Reihenfolge angezeigt werden.

3. Benutzerkontrolle und Zufriedenheit
Der Benutzer macht bei der Bedienung des Systems, d. h., bei der Auswahl von Systemfunktionen, Fehler. Dann braucht er einen deutlich markierten Notausgang, um die unerwünschte Auswahl rückgängig machen zu können. Dies sollte er können, ohne aufwendige Dialoge absolvieren zu müssen. Funktionen wie Rückgängig und Wiederholen sollten von einer Software unterstützt werden.

4. Konsistenz und Standards
Der Benutzer sollte nicht über unterschiedliche Terminologien, Situationen oder Aktionen, welche eigentlich dieselbe Sache beschreiben, stolpern. Eine Anwendung sollte deshalb den Plattformkonventionen folgen und gleiche Sachverhalte einheitlich darstellen.

5. Fehlerprävention
Ein umsichtiges Design, welches Fehlern vorbeugt, bevor sie auftreten können, ist immer besser als eine gute Fehlermeldung. Deshalb sollten Mehrdeutigkeit und Unübersichtlichkeit vermieden werden sowie unwichtige Informationen verborgen bleiben.

6. Erkennen ist besser als Erinnern
Objekte, Aktionen und Optionen sollten sichtbar sein, d. h., auf der Oberfläche verfügbar sein. Der Benutzer sollte sich nicht an jede Dialogreihenfolge erinnern müssen, wie er Instruktionen für das Benutzen des Systems erreichen kann. Sie sollten erreichbar sein, wann immer sie gebraucht werden.

7. Flexibilität und Effizienz
Beschleunigungen oder Abkürzungen (zum Beispiel Icon in Toolbars oder Tastaturkürzel), welche vom unerfahrenen Benutzer nicht gesehen und nicht gebraucht werden, können häufig die Geschwindigkeit der Benutzung für erfahrene Benutzer erhöhen. Der Benutzer sollte diese zudem selbst gestalten können.

8. Ästhetik und minimales Design
Dialoge sollten keine Informationen enthalten, welche unwichtig oder ungewöhnlich sind. Jede zusätzliche Information in einem Dialog steht im Wettbewerb mit den relevanten Informationen und vermindert ihre relative Sichtbarkeit. Informationen sollten stets so kurz und gehaltvoll wie möglich sein, d. h., jegliche unnötige Information sollte vermieden werden.

9. Hilfe für den Anwender beim Erkennen, Diagnostizieren und Rückgängig machen von Fehlern
Fehlermeldungen sollten in einfacher Sprache erfolgen. Es sollten keine Codes oder dem Benutzer unverständliche Meldungen ausgegeben werden. Sie sollten die Ursache des Problems präzise beschreiben und konstruktiv einen Lösungsweg aufzeigen. Der einfachste Lösungsweg ist das Rückgängig machen von Fehlern.

10. Hilfe und Dokumentation
In jedem Fall ist es besser, wenn das System ohne Dokumentation verwendet werden kann. Es kann Hilfe und Dokumentation angeboten werden. In diesem Fall sollte jede Information leicht zu suchen und zu finden sein. Die Hilfe sollte auf die Aufgabe, welche der Benutzer gerade bewältigt, fokussiert sein. Es sollte eine konkrete Liste von Schritten ausgegeben werden, welche einen Lösungsweg für die Aufgabe beschreibt und den Benutzer möglichst schnell zum Ziel führt.

4.9 Gestaltung von Benutzeranleitungen

Handbücher für Software erfüllen zwei Funktionen, zum einen wird die Software selbst dokumentiert und zum anderen sind es Nachschlagewerke für Benutzer. Wenn die Chance besteht, sollte man die Benutzeranleitung parallel zur Software-Entwicklung laufen lassen. Denn alles, was die Autoren nicht gut beschreiben können, führt später zu Benutzungsproblemen. Die Benutzerdokumentation muss laut DIN EN ISO 12119 folgende Kriterien erfüllen: Vollständigkeit, Richtigkeit, Widerspruchsfreiheit, Verständlichkeit und Übersichtlichkeit.

Handbücher für Benutzer sollten die in der Tabelle benannten Kriterien erfüllen:[15]

Für eine offizielle Software-Ergonomie-Prüfung wird die DIN EN ISO 12119[16] herangezogen. Die DIN EN ISO 12119 definiert einige softwaretechnische Eigenschaften von Softwareprodukten, die als Voraussetzung zur Erreichung der Nutzungsziele gegeben sein müssen. Dazu gehört u. a. auch die Produktbeschreibung. Software-Handbücher tragen wesentlich zur Benutzbarkeit eines Software-Produktes und zur Motivation des Benutzers bei.[17]
Gute Handbüche für Software zu schreiben, ist nicht einfach und erfordert in der Regel eine Ausbildung als „Technischer Dokumentar".

Kriterium	Beschreibung, Beispiel	ISO 12119 (sinngemäß)
Aufgaben-bezogenheit	An Aufgaben anstatt an funktionalen Möglichkeiten der Software orientiert. Statt die Funktionen einzeln zu erläutern, soll die Beschreibung anhand der Arbeitsaufgaben erfolgen.	Die Produktbeschreibung muss außerdem den wesentlichen Leistungsumfang in Form von unterstützten Arbeitsaufgaben und Tätigkeiten sowie deren Ergebnissen umfassen, die mit dem Produkt erzielbar sind. Die Funktionsübersicht sollte aus Benutzersicht nicht in Form von Funktionsaufzählungen formuliert sein. Die Benutzerdokumentation muss die wesentlichen Arbeitsaufgaben und die zu ihrer Durchführung erforderlichen Arbeitsschritte beschreiben und handlungsleitende Informationen geben (d.h. erläutern), wie diese Schritte mit dem Produkt durchzuführen sind.
Benutzer-bezogenheit	Für Anfänger und geübte Benutzer geeignet. Bereits im Inhaltsverzeichnis finden sich Hinweise, was für die jeweilige Gruppe geeignet ist.	Die zur Durchführung der Arbeitsaufgaben erforderlichen Vorkenntnisse der Benutzer sind zu benennen. Die Benutzerdokumentation muss für die vorgesehene Zielgruppe im Zusammenhang mit der durchzuführenden Arbeitsaufgabe verständlich sein.
Fehler-orientierung	Es finden sich Hinweise zu Fehlerbehebung (wie können gemachte Fehler behoben werden). Fehlerannullierung (wie können Eingaben rückgängig gemacht werden), Fehlermeldungen.	Benutzerhandlungen dürfen nicht zu undefinierten Zuständen der Software oder zu Datenverlust (ohne Warnung, die vom Benutzer bestätigt wurde) führen. Dies gilt auch dann, wenn Kapazitätsgrenzen der Software erreicht oder überschritten werden oder der Benutzer unerlaubte Eingaben/Handlungen vornimmt (Ursachen/Auswirkungen, die nicht im Einflussbereich der Software liegen, sind davon ausgenommen). Außerdem ist auf Randbedingungen bzw. Grenzwerte des Programms hinzuweisen, soweit diese für die Arbeitsaufgaben relevant sein können.
Modulare Darstellung	Jedes Kapitel ist für sich verständlich. Dokumentation ist auch als Nachschlagewerk zu benutzen. Querverweise zu anderen Kapiteln sind möglichst gering halten (wenn, dann z. B. „Weiter in Kapitel X für den geübten Benutzer"). Handbuch muss auch auszugsweise – nicht nur von Anfang an – gelesen werden können.	Hier gibt es keine entsprechenden Forderungen der ISO 12119.

[15] Richenhagen, Prümper, Wagner: Handbuch der Bildschirmarbeit, 1997
[16] ISO 12119 (1995) Informationstechnik. Software-Pakete. Qualitätsanforderungen und Prüfung
[17] Boedicker, D.: Handbuch-Knigge, 1990 Wissenschaftsverlag

Kriterium	Beschreibung, Beispiel	ISO 12119 (sinngemäß)
Seitenreduktion	Weniger ist mehr: Seitenzahl und Kapitelumfang begrenzen. Kein Abschnitt und Beschreibung einer wesentlichen Aktion länger als 3 Seiten. Umfangreiche Dokumentationen sollten durch „Referenzkarten“ ergänzt werden.	Hier gibt es keine entsprechenden Forderungen der ISO 12119.
Transparenz	Das Dokument soll eine ganzheitliche, innere Vorstellung des Programms ermöglichen. Dafür geeignet sind: • Grafische Darstellungen (Menüs oder Masken als hardcopys) • Orientierungsdiagramme aufgabenorientiert, piktografisch • Repräsentation der Struktur des Programms • Texte durch bildliche Darstellungen, einprägsame Symbole ergänzen bzw. ersetzen. • Lesefreundliche Darstellung, wie Zusammenfassung in Tabellenform. • Farbliche Gestaltung, übersichtliches Layout, klares Schriftbild. • Eine ansprechende grafische Gestaltung (Schriftart, Symbole, Gliederung) unterstützt die Verständlichkeit wesentlich.	Die Benutzerdokumentation muss übersichtlich strukturiert sein (z. B. Inhaltsverzeichnis) und schnelles Suchen ermöglichen (z. B. Stichwortverzeichnis, Stichwortsuche). Zusammenhänge müssen beim Lesen gut erkennbar sein (z. B. durch geeignete grafische Gestaltung, Screenshots, Verweise).
Handhabung	Handbücher sollten aufgeschlagen liegen bleiben können! Kopf- und/oder Fußzeile mit Kapitelbezeichnungen	Die Benutzerdokumentation muss übersichtlich strukturiert sein (z. B. Inhaltsverzeichnis) und schnelles Suchen ermöglichen (z. B. Stichwortverzeichnis, Stichwortsuche). Zusammenhänge müssen beim Lesen gut erkennbar sein (z. B. durch geeignete graphische Gestaltung, Screenshots, Verweise).
Überblick & Register	Inhaltsverzeichnis, Stichwortregister, Glossar, Liste der Fehlermeldungen	Hier gibt es keine entsprechenden Forderungen der ISO 12119.
Verständliche Sprache	Einheitliche, verständliche Wortwahl. Handlungsauffordernde, höfliche Formulierungen (statt „Sie müssen“ besser „wählen Sie“). Die Verständlichkeit von Texten wird gefördert durch: Anordnung der Texte und/oder bildliche Darstellungen in der Reihenfolge der Handhabung bei erstmaligem Benutzen (erwarteter Umgang). Abfassen der Abschnitte in der Reihenfolge des Denkablaufs: siehe-denke-benutze. Klares Gliedern der Abschnitte sowie problem- und handlungsbezogene Wahl der Überschriften. Aufgabenbezogene Formulierungen, z. B. „Wie lösche ich Text“ statt „Grundfunktionen des xy-Programms“. Kurzes und prägnantes Formulieren/Beschränken des Textes auf das Wesentliche. Erklären von Fachausdrücken, Symbolen und stilisierten Darstellungen. Vermeiden unterschiedlicher Benennungen für denselben Gegenstand/Sachverhalt. Unmissverständliche Zuordnung der bildlichen Darstellung zum Text.	Hier gibt es keine entsprechenden Forderungen der ISO 12119.

5



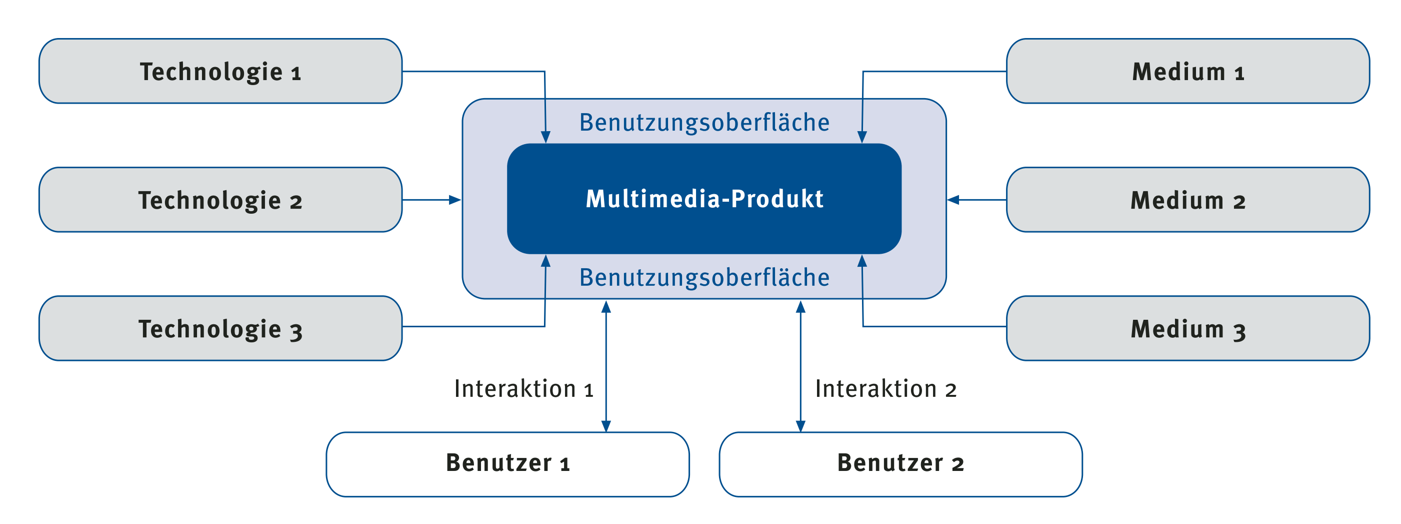
5 Software-Ergonomie für Multimedia

5.1 Was sind Multimedia-Systeme?

Unter Multimedia-System wird ein Produkt verstanden, das eine Mischung aus verschiedenen Technologien und Medien ist. Unter Technologien versteht man in diesem Zusammenhang, z. B. in welcher Sprache die Anwendung programmiert wird (HTML, Java...) oder welche Datenbank eingesetzt wird. Von den Medien ist in der Regel mindestens eines ein dynamisches Medium.
In der ISO 14915[1] wird Multimedia als eine Kombination von statischen und/oder dynamischen[2] Medien verstanden, welche interaktiv gesteuert und simultan in einer Anwendung präsentiert werden können. Medien sind z. B. Text, Grafik, Fotos, Videos, Animation, Geräusche, Musik, Sprache.
Multimedia umfasst also:

- Einsatz mehrerer Medien, integrierte Nutzung
- Kombination von statischen und/oder dynamischen Medien
- Interaktiv steuerbar
- Simultan in einer Applikation präsentierbar
- Integrierte Nutzung zur Erreichung des gleichen Zwecks

Im Arbeitsumfeld verbreitete IT-Systeme erlauben Benutzereingaben bisher überwiegend über Text und Grafik, d. h. über Menüdialoge, direkte Manipulation (z. B. mit der Maus das Dokument in den Papierkorb ziehen), Formulare oder vereinzelt noch über Kommandodialoge (z. B. del = für löschen).
Ausgaben der Rechner (für die Benutzer) erfolgten nur visuell über den Bildschirm.
Die Entwicklung geht dahin, natürliche Interaktionsformen des Menschen (Sprache, Gestik) für den Rechner erkennbar zu machen und andererseits die Ausgabearten der Rechner im Bereich Akustik, Haptik usw. zu entwicklen. Denn Tastatur und Maus sind nur Hilfsmittel zur Verständigung mit dem Rechner. Sicher für bestimmte Zwecke gut geeignet, für andere Zwecke weniger.
Erste beeindruckende Entwicklungen gibt es im Freizeit/Unterhaltungsbereich (Computerspiele u. ä.).
Hier sollen jedoch nur Multimedia-Systeme aus dem Arbeitsbereich und hier speziell die Informations- und Wissensvermittlung betrachtet werden. Also z. B. e-learning-Software/CD-ROMs, elektronische Nachschlagewerke, Sprachlernprogramme usw.
Darüber hinaus sind natürlich auch z. B. Navigationssysteme in Autos, Web-Portale, Mobiltelefone oder Anwendungen in der medizinischen Informatik multimediale Anwendungen.
Auch wird der zunehmende Einsatz mobiler Geräte im Arbeitsbereich zukünftig stärker zu beobachten sein. Ob dies mobile Erfassungsgeräte im Pflegebereich, für Warenbestände oder Verbrauchsdaten sind.



Quelle: K. Wunderlich: Media Engineering Veranstaltung „Informatica feminale" 2000

[1] ISO 14915 Software Ergonomie für Multimedia-Benutzungsschnittstellen
[2] statisch: z. B. ein einzelnes Bild, dynamisch: z. B. ein Abfolge von Bildern: Video; Sound...



Multimedia-Oberflächen

Benutzungsoberfläche eines nautischen Notfall- und Navigationssystems (NAVACS) [4]

Peter Gabriel: Multimedia-CD „Eve“

Zu sehen sind die Auslöseelemente für „Load“, „Quit“ und „Begin“, zu denen man akustisch und visuell geleitet wird.



5.2 Voraussetzungen des Menschen und Gestaltung von Multimedia-Software

5.2.1 Die menschlichen Sinne

Aus Sicht der Nutzer können Medien in solche, die visuell wahrgenommen werden, wie Text, Grafik, Foto, Video oder Animation, unterteilt werden und Medien, die akustisch wahrgenommen werden, wie Geräusche, Musik, Sprache. Der visuelle und der akustische Sinn des Menschen sind die beiden am stärksten genutzten menschlichen Sinne bei Multimedia-Anwendungen. Üblicherweise geht man von fünf menschlichen Sinnen aus.
Es gibt weitere menschliche Sinne, wie den Gleichgewichtssinn, Temperatursinn usw., die hier nicht betrachtet werden. Einige werden sicher für zukünftige Mensch-Computer-Schnittstellen eine Bedeutung haben. Der Gleichgewichtssinn ist heute schon in sog. Virtual Reality[3]-Anwendungen von Bedeutung. Bekannt ist die Pilotenausbildung in Flugsimulatoren. Ebenfalls in VR-Systemen wird durch eine Kraftrückkoppelung (Tastsinn) Medizinstudenten das richtige Spritzen beigebracht, d.h. sie erfahren, wie die Nadel durch die Haut dringt und auf Blutbahnen, Sehnen und Knochen trifft.

Visueller Sinn

Ergonomisch betrachtet, ist die stärkere Nutzung des visuellen Systems des Menschen naheliegend, da dieses System sehr leistungsfähig ist. Ca. 30 % der Gehirnaktivität ist für das Sehen ausgelastet. Wir können gleichzeitig viele optische Informationen aufnehmen.

Menschliche Orientierung, ob in Software oder im „normalen Leben“, läuft fast ausschließlich visuell ab. Man spricht auch vom Menschen als Augentier.
Spielt dieser Sinn schon für herkömmliche Software eine große Rolle, so wird er für multimediale Software noch stärker in Anspruch genommen. Dies kann einerseits der menschlichen Wahrnehmung entgegenkommen, da der Mensch z. B. sich kontinuierlich ändernde Informationen besser aufnehmen kann, als statische Informationen. Auch kann der Mensch selektiv wahrnehmen, d. h., man sieht nur, was man sehen will.

Sinn / Organ	Nutzung für Multimedia-Software
visueller Sinn / Auge	Visualisierung von Abläufen, die real nicht sichtbar sind (Blutfluss in Adern)
auditiver Sinn / Ohr	Einsatz verschiedener Stimmen zur Verdeutlichung von Unterschieden
Tastsinn (Haptik) / Haut	Blinde ertasten über eine Brailletastatur Texte, Touch-Screen-Bildschirme
Geruchssinn / Nase	erste Prototypen bisher keine praktische Relevanz
Geschmackssinn / Zunge, Nase	bisher keine praktische Relevanz

[3] Virtual Reality: Als Virtuelle Realität wird die Darstellung und gleichzeitige Wahrnehmung der Wirklichkeit und ihrer physikalischen Eigenschaften in einer in Echtzeit computergenerierten virtuellen Umgebung bezeichnet.

[4] von CD „AVECS“ Company and ISM Products Presentation © 1994–97, AVECS GmbH, Potsdam

Zu beachten ist die zeitliche Auflösung des visuellen Sinns. Sie beträgt 15–50 Millisekunden, das heißt der Reiz muss so lange vorhanden sein, um wahrgenommen zu werden.
Für Multimedia-Anwendungen speziell zu berücksichtigen ist die menschliche Fähigkeit zur Wahrnehmung von Bewegung. Sehen wir Einzelbilder in zu langsamer Abfolge, nehmen wir nur eine ruckende Bewegung wahr, wie z. B. bei unzureichender Daten-Leitung für eine Videokonferenz. Man benötigt 20–25 stehende Bilder pro Sekunde, um einen fließenden Bewegungseindruck zu erzeugen.
Die Farbwahrnehmung wurde bereits in vorhergehenden Kapiteln behandelt.

Auditiver Sinn

Der visuelle Sinn ist zwar der vorrangige Sinn, der von der Umwelt auch am offensichtlichsten angesprochen wird. Zur Orientierung werden aber die übrigen Sinne auch benutzt, wenn auch unbewusst, wie z. B. der leistungsstarke akustische Sinn.
Der auditive Sinn ist unser sekundäres, unterstützendes Wahrnehmungssystem, welches eine zunehmende Bedeutung für die Mensch-Computer-Interaktion hat.
Um akustische Informationen zu erhalten, muss Schall zunächst dekodiert werden. Die akustischen Eindrücke werden vom Ohr an das Gehirn übermittelt und dort verarbeitet.
Die Information erschließt sich über das auditive Gedächtnis bzw. die Kombination von konventionalisierten und erlernten Geräuschen bzw. Tönen. Es können also bekannte Geräusche gefiltert bzw. schnell aufgenommen werden, die mit Hilfe des Gedächtnisses in einen Kontext eingebunden werden können. Je vertrauter ein Geräusch, desto schneller kann es verarbeitet werden.
Akustische Reize können optische Informationen unterstützen, einprägsamer und authentischer machen und in manchen Fällen sogar besser übermitteln, z. B. dann, wenn Information nicht (oder nicht vollständig) visuell dargestellt werden kann.
Ein einfaches Beispiel ist das Läuten der Türglocke oder das Klicken der Blindenampel.
Rückmeldungen über Systemreaktionen an Benutzer können auch visuell und akustisch codiert werden. Eine typische optische Rückmeldung ist bei einem klickbaren Button z. B. das „highlighten“ (optisches Hervorheben). Eine typische zusätzliche akustische Rückmeldung beim Drücken dieses Buttons wäre ein Klickgeräusch. Bei den meisten Mobiltelefonen kann man diese Funktion ein- bzw. abstellen.
Akustische Wahrnehmung ist immer zeitabhängig. Wenn z. B. bei Sprachapplikationen, wie sie für viele Arten von Service über Telefon eingesetzt werden (wählen Sie die 1 um, … , wählen Sie die 2 um, …). Wenn diese Aufzählung zu lang oder umständlich formuliert ist, weiß der Anrufer am Ende nicht mehr, was er auswählen soll.

Die zeitliche Auflösung des akustischen Sinns beträgt 2–3 ms, das heißt akustische Signale werden schneller wahrgenommen als optische Signale. Warnmeldungen, gerade in sicherheitskritischen Anwendungen, werden in der Regel akustisch gemeldet.

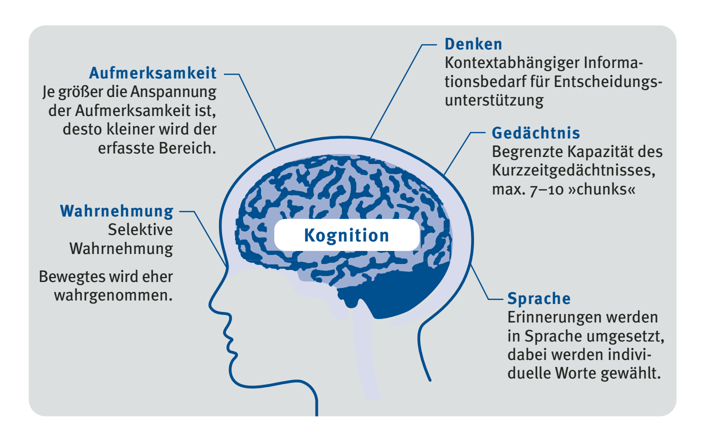
Dass wir kurze Sounds/Geräusche natürlichen bzw. erlernten Ursprungs sofort aufnehmen, ohne sie lange dekodieren zu müssen, hat einen biologischen Sinn, nämlich Schutz bzw. Voraussetzung zum Überleben. (Ein Baum, der umfällt, gibt ein typisches Geräusch von sich, ebenso wie ein nahender LKW.)
Durch die Einbeziehung auditiver Informationen lassen sich ergänzende Informationen mit Hilfe eines zusätzlichen Sinnes (zusätzlich zu dem visuellen) darstellen. Der Bildschirm kann dabei von Text entlastet werden und es können gleichzeitig Informationen visuell und auditiv verarbeitet werden. Außerdem können z. B. Bilder durch Originalgeräusche einprägsamer und authentischer dargestellt werden.
Berücksichtigt werden muss aber, dass eine akustische Rückmeldung die Aktion des Benutzers unterstützen und nicht verwirren soll, d. h. der Sound sollte logisch oder erwartet/nachvollziehbar sein. Denn ein wesentlicher Unterschied zum visuellen System ist, dass kaum Schutzmechanismen bestehen.

Möglichkeiten und Grenzen

Die Nutzung verschiedener Medien (z. B. ein Video, welches durch einen gesprochenen Text kommentiert wird) kommt den menschlichen Kommunikations-Fähigkeiten ebenfalls entgegen, denn wir nehmen auch Informationen im Alltag über mehrere Sinneskanäle auf. Es besteht allerdings die Gefahr, dass es durch unkoordinierten Medieneinsatz zu einer kognitiven Mehrbelastung und damit zu einer Überforderung der menschlichen Aufnahmefähigkeit kommt.
Bei der Herstellung einer e-learning-Software, bei der mit Text, Bildern, Videosequenzen und Sprachausgabe gestaltet wird, müssen ergonomische Gestaltungsrichtlinien berücksichtigt werden. Das heißt: Graphik, Sprache und Video sollen nicht exzessiv eingesetzt werden, wie es bei PC-Abenteuerspielen beabsichtigt sein kann.
Aspekte wie Didaktik, Regie, Komposition, Dramaturgie und Navigation sollten berücksichtigt werden, um Medien für die menschliche Wahrnehmung sinnvoll einzusetzen und aufeinander abzustimmen.

5.2.2 Kognitive und psychologische Faktoren

Neben den menschlichen Sinnen gibt es weitere kognitive und psychologische Faktoren, die für die Multimedia-Gestaltung berücksichtigt werden sollen. Ursprünglich wurden diese Faktoren im Rahmen der Gestaltung sicherheitskritischer Software-Systeme beschrieben. Forderungen, die für sicherheitskritische Systeme formuliert wurden, lassen sich auch für Multimedia-Produkte anwenden.



Kognitive Faktoren

Unter Kognition versteht man den Prozess des Erkennens. Kognition umfasst den Prozess des Wahrnehmens und Verarbeitens von Informationen bis zum Umsetzen in Handlungen mit den dabei im Menschen wirkenden Bedingungen psychischer, physiologischer und sprachlicher Art.
Konkret geht es dabei um

- Informationsaufnahme
- Behalten und Erinnern
- Verständnis von Sprache
- Erkennen von Figuren und Zeichen
- Entscheidung zwischen Alternativen und andere Problemlösungsprozesse

Als Faktoren mit übergreifenden Funktionen im kognitiven Bereich gelten das Bewusstsein, die willkürliche und die unwillkürliche Aufmerksamkeit und das Gedächtnis. Weitere kognitive Faktoren sind die Wahrnehmung, das Denken und das Handeln.

Eine allgemein anerkannte präzise Definition von Bewusstsein liegt nicht vor. **Bewusstsein** beschreibt die Fähigkeit, sich kraft Beobachtung, Urteil und Verhalten im Kontrast zu seiner Umwelt zu erleben und sie zu beeinflussen. Es ist ferner die Instanz, in der mentale Zustände wie beispielsweise Schmerz, Wut und Farbempfindung repräsentiert werden. Das Bewusstsein steuert keineswegs ausschließlich das Verhalten, denn dieses geschieht auch unbewusst. Bewusstsein ist eng mit der Fähigkeit des Erinnerns verbunden.

Aufmerksamkeit ist der zentrale kognitive Faktor, der die Ausrichtung des Konzentrations-, Klarheits- und Intensitätsschwerpunktes auf Gegenstände der Wahrnehmung, des Denkens oder Handelns willkürlich oder unwillkürlich lenkt. Dadurch erfolgt eine Selektion, der ins Bewusstsein gelangenden Informationen, so dass die nicht erfassten unbemerkt bleiben können.

Beim **Gedächtnis** handelt es sich um einen Speicher im kognitiven Bereich für Erfahrenes, Erlerntes und Erdachtes, mit der Fähigkeit, bei Bedarf dem Bewusstsein das zu reproduzieren, was für die jeweilige Situation benötigt wird.

Denken und **Wahrnehmung** beeinflussen sich gegenseitig. Hierbei handelt es sich um einen psychophysischer Prozess, bei dem physikalische Reize (Lichtstrahlen, Töne, Vibrationen) über Organe (z. B. Auge, Ohr, Tastsinn) aufgenommen werden und bewusstseinsfähig gemacht werden.

Beim **Denken** handelt es sich um einen wesentlichen Vorgang im kognitiven Bereich für das Lösen von Aufgaben. Grundfunktionen sind das Erzeugen von Vorstellungen und Begriffen, Erfassen, Bestimmen und Ordnen ihrer Beziehungen, das Vergleichen bzw. Unterscheiden, das Urteilen und Schlussfolgern. Spontane Intention des Denkens ist es, größtmögliche Klarheit, Einfachheit, Übersicht und Einsicht in Bezug auf Tatsachen, Vorgänge, Ziele und Probleme zu erhalten. Entscheidende Stärke des menschlichen Denkens ist die rasche Ein- und Umstellfähigkeit, die es erlaubt, auch unerwartete Situationen mit ad hoc entwickelten Handlungsstrategien zu meistern. Denken und Handeln sind eng miteinander verbunden.

Handeln: Analysen der Ziele, das Generieren von Handlungen und die Handlungskontrolle stehen im Vordergrund. Die Fähigkeit, parallel zu handeln, ist bei Menschen begrenzt.

Faktor	Bedeutung für Multimedia
Bewusstsein	Im Durchschnitt werden nicht mehr als 3 Informationseinheiten gleichzeitig im Bewusstsein präsent gehalten. Veränderungen des Bestandes im Bewusstsein können nur einzeln, also sequentiell vorgenommen werden.
Aufmerksamkeit	Je größer die Anspannung der Aufmerksamkeit ist, desto kleiner wird der erfasste Bereich. Die unwillkürliche Aufmerksamkeitshaltung (ohne absichtliches Zutun des Menschen) herrscht beim Menschen üblicherweise vor und sollte deshalb bei der Überwachung zeitkritischer Prozesse weitgehend zum Tragen kommen.
Gedächtnis	Der Mensch erinnert sich besser an Dinge, die über mehrere Sinneskanäle aufgenommen wurden.
Wahrnehmung	Es gibt Dinge, die sehen wir, nehmen sie aber nicht bewusst wahr. In Versuchen wurde nachgewiesen, dass kurzes Einblenden (unterhalb der Wahrnehmungsschwelle) eines Bildes in ein laufendes Video Spuren im Gehirn hinterlässt, der Mensch sich jedoch nicht an dieses Bild erinnert.
Denken	Spontane Intention des Denkens ist größtmögliche Einfachheit bei einer Problemlösung. Dieser Intention kann absichtlich gegengesteuert werden (beim Lösen von Rätseln: „um die Ecke denken").
Handeln	Zeitbedarfe für das Ausführen von Handlungen müssen berücksichtigt werden.

Gestaltungshinweise, die sich aus der Berücksichtigung kognitiver Faktoren ergeben, sind:

- Bewegtes wird eher wahrgenommen
- markante Strukturen führen den Blick
- akustische Signalmeldung für kritische Situationen verwenden
- Informationen müssen solange angezeigt werden, bis sie vom Nutzer quittiert werden

Die hier benannten kognitiven Faktoren stehen in Zusammenhang mit der oben beschriebenen Sinneswahrnehmung. So wird z. B. je nach Erwartungshaltung die Aufmerksamkeit zu einem Sinneskanal gelenkt, der dann auch schneller verarbeitet.

Durch die kognitiven Faktoren begründet sind weitere Empfehlungen für Anwendungen im sicherheitskritischen Bereich, die jedoch für multimediale Anwendungen durchaus übernommen werden sollten (Auszug[5]):

- Empfehlungen passender Codierungsarten für bestimmte Aufgaben (Ablesen qualitativer Werte = analoge Darstellung, Ablesen quantitativer Werte = digitale Darstellung)
- Bewegungseffektstereotypien (Drehbewegung nach rechts = Einschalten, Drehbewegung nach links= Abschalten)
- Maßnahmen gegen Über– und Unterforderungssituationen (Vigilanz[6])
- Dimensionalität und Kompatibilität beachten (Räumlichkeit, Seitigkeit)

[5] Näheres hierzu kann bei der Autorin der Broschüre nachgefragt werden.

[6] Vigilanz: Wachsamkeit, Fähigkeit, über längere Zeit trotz reizarmer Umgebung und Ermüdung wachsam zu bleiben; Bereitschaft, auf unerwartete Ereignisse angemessen zu reagieren.

5.3 Gestaltung von Multimedia-Produkten

Hier geht es ausschließlich um die ergonomische Gestaltung von Multimedia-Produkten und nicht um ästhetisches Erscheinen o. ä. Auch in der Norm ISO 14915 ist ausdrücklich betont, dass Design im herkömmlichen Sinn nicht Bestandteil der Norm ist.
Redundante (mehrfache) Informationsaufnahme bei Multimedia-Anwendungen wird i. d. R. nicht als störend empfunden, sondern fördert und beschleunigt die Wahrnehmung. Wichtig ist eine gute Synchronisation der Medien. Eine um Sekunden verschobene Tonspur eines Film wirkt sehr störend.
Andererseits kann die Nutzung von Multimedia-Software zu einer hohen Wahrnehmungsbelastung führen, die strukturelle und semantische Komplexität und die potentielle Informationsüberflutung zu kognitiven Mehrbelastungen führen.

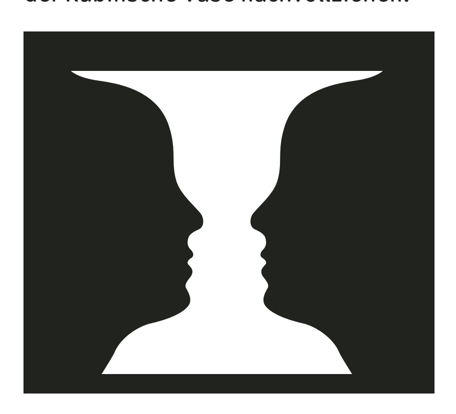
5.3.1 Visuelle Gestaltung

Neben den allgemein bereits benannten Grundlagen und Richtlinien zur Informationsdarstellung gibt es einige spezifische bzw. für Multimedia besonders zu beachtende Erkenntnisse über die Psychologie der visuellen Wahrnehmung.

Figur/Grundtrennung
Bereits im Kapitel 3 ist von den Gestaltgesetzen die Rede. Bei multimedialer Software spielt die Gestaltung des Bildschirms in flächige Bereiche eine größere Rolle als bei Büroanwendungen. Z. B. ist vorstellbar, dass in e-learning-Software Tafeln, auf denen etwas steht, als Hintergrund gezeigt werden. Dafür kann z. B. die Erkenntnis der Gestaltpsychologen[7] interessant sein, dass

- eher helle, symmetrische, konvexe oder kleine Flächen zur Figur werden,
- dunkle, asymmetrische, konkave oder größere Flächen zum Hintergrund werden.

Dieses Phänomen kann man an dem bekannten Vexierbild[8] der Rubinsche Vase nachvollziehen.



Vexierbild: Rubinsche Vase

Objektwahrnehmung
Das Erkennen von Oberflächen wird aus Konturen und Texturen abgeleitet. Dabei können Objekte gesehen werden, die in der Realität räumlich gar nicht vorkommen. Unterschiede in der Textur führen zum Erkennen von unterschiedlichen Flächen durch Entstehung von Konturen an den Grenzen zweier Texturen.

Wahrnehmung räumlicher Tiefe
Die Wahrnehmung von Tiefe ist durch das stereoskopische Sehen möglich.

Die Darstellung am Bildschirm erfolgt durch Verdeckung von Konturen, durch unterschiedliche Schattierung und Sättigungsgrade.

[7] Wertheimer 1923
[8] Vexierbild ist ein Rätselbild, bei dem z. B. eine Figur oder ein Gegenstand zu suchen ist. Meist ist die zweite Darstellung nicht einfach zu erkennen.

5.3.2 Verwendung von Metaphern

Metaphern sind Repräsentationen der realen Welt. Die in herkömmlicher Software verwandte „Schreibtisch-Metapher" passt gut für Anwendungen im Büroverwaltungsbereich, jedoch weniger für multimediale Anwendungen. Denkbar sind hierfür e-learning-Programme z. B. die Klassenraum-Metapher oder für Nachschlagewerke die Buchmetapher. Wichtig ist dabei, dass die Metapher vom Benutzer so genutzt werden kann, wie er es in der realen Welt auch tun würde. Dies bedeutet z. B. dass man bei der Metapher „Buch" auch hin- und herblättern kann.

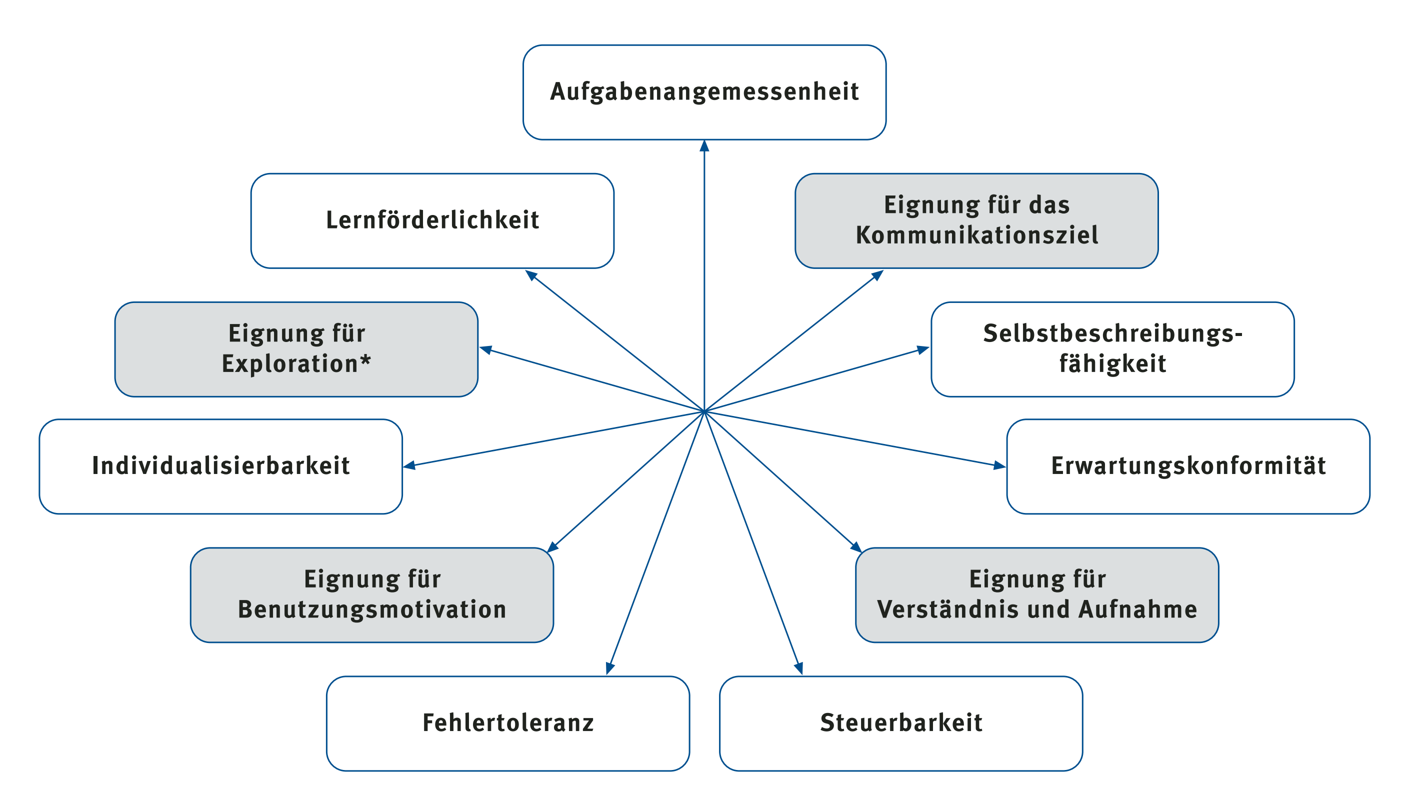
5.3.3 Dialoggestaltung

In der ISO 14915[9] finden sich Regeln zur Gestaltung von Multimedia-Produkten, die sich eng an die Norm zur Dialoggestaltung (ISO 9241) anlehnen und diese ergänzen.

Kriterium	**Beispiel in Multimedia-Anwendung**
Aufgabenangemessenheit	Ein e-Learning-Programm für Musikinstrumente zeigt ein Video, in dem man die Handbewegungen sehen kann.
Selbstbeschreibungsfähigkeit	Beim Bewegen des Cursors über Hotspots einer Applikation wird die Beschreibung der Funktion angezeigt.
Steuerbarkeit	Die Lautstärkeregelung kann vom Benutzer an- und ausgeschaltet werden.
Erwartungskonformität	Steuerungsanzeigen für Abspielen und Stoppen eines Mediums wirken in der gesamten Applikation gleich.
Fehlerrobustheit	Wird ein Video unbeabsichtigt gestoppt, kann an der Position wieder gestartet werden, an dem es gestoppt wurde.
Individualisierbarkeit	Der Benutzer kann Präferenzen einstellen, wie das Ausgabemedium bestimmen, Bookmarks setzen.
Erlernbarkeit	Eine visuelle Anzeige der Navigationsstruktur ist vorhanden.

Beispiele für die Dialogkriterien angewendet auf Multimedia-Systeme

[9] DIN EN ISO 14915 Software-Ergonomie für Multimedia-Benutzungsschnittstellen



Ergänzende Dialogkriterien für Multimedia-System

* Exploration: durch „Ausprobieren“ erkunden

Speziell für Multimedia-Systeme werden weitere vier Kriterien benannt, die man z. B. an der Gestaltung von e-learning-Software nachvollziehen kann.

Eignung für das Kommunikationsziel

Die Applikation ist so gestaltet, dass die Ziele des Anbieters der Information und gleichzeitig die Ziele oder Aufgaben des Benutzers oder Empfängers der Information getroffen werden.

Dies bedeutet, dass in einer Anwendung die Absichten zwei verschiedener Nutzergruppen berücksichtigt werden müssen. Im e-learning-Bereich kann dies das Interesse der Lehrenden sein, den Lehrstoff unterhaltsam anzubieten und das Interesse des Lernenden, den Stoff in für ihn passenden Einheiten abzurufen.

In der Norm stehen konkretisierte Beispiele wie:

- Überzeugende Argumente können mehrfach dargeboten werden, um die Schlüsselbotschaft „rüberzubringen“.

Eignung für Wahrnehmung und Verständnis

Die Software soll so gestaltet sein, dass die zu übermittelnden Informationen leicht erfasst und verstanden werden können. Hierbei sollen die auch in der DIN EN ISO 9241-12 (Informationsdarstellung) bereits beschriebenen Kriterien beachtet werden:
Klarheit, Entdeckbarkeit, Unterscheidbarkeit, Lesbarkeit, Kürze und Konsistenz. Diese sind für Multimedia-Anwendungen besonders wichtig, da die Informationsdarstellung komplex und flüchtig ist.

Zusätzlich sollte folgendes beachtet werden:

- Vermeiden von einer Überlastung der Wahrnehmung
- Vermeiden von Informationsüberladung, die durch zeitbegrenzte Darstellung verursacht ist (z. B. Newsticker)
- Vermeiden von Überlastung durch zu viele gleichzeitige Aktivitäten
- Berücksichtigen von Wahrnehmungsdifferenzen (Accessibility)[10]
- Unterstützen des Benutzerverständnisses

[10] Accessibility = Zugänglichkeit für Nutzer mit körperlichen Beeinträchtigungen

Eignung für Erforschung/Exploration
Eine Multimedia-Applikation ist für die Erforschung geeignet, wenn der Benutzer relevante oder interessante Informationen findet, mit wenig oder keinen Kenntnissen über die Art, Ausmaß und Struktur der Information. Dies kann erreicht werden durch:

- Unterstützen der Benutzerorientierung
- Unterstützen einer transparenten Navigation
- Anbieten alternativer Navigationspfade
- Information strukturieren
- Erleichterung der Rückkehr zu signifikanten Punkten, z. B. durch Visualisierung des Pfades
- Anbieten von Suche und Navigationshilfen, z. B. eine Sitemap ist vorhanden
- ggf. verschiedene Medien zur Auswahl für den Benutzer anbieten

Eignung für Benutzungsmotivation
Wenn es für die Aufgabe geeignet ist, sollte die Applikation so gestaltet sein, dass der Benutzer sich engagieren kann, z. B. um die Aufmerksamkeit zu erhalten und zu motivieren. Das könnte z. B. für ein e-learning-Programm bedeuten, dass der Benutzer bei erfolgreichem Lösen einer besonders kniffeligen Aufgaben mit einer lustigen Animation „belohnt“ wird. Oder es wird ein hoher Grad an Realität in einer Simulation geboten – in Kombination mit einem hohen Grad an Interaktivität unterstützt dies ein Benutzerengagement.

5.3.4 Inhalte und Navigation[11]

Bei Multimedia-Software spielt die Steuerung der einzelnen Medien und die Navigation eine wichtige Rolle. Multimedia-Steuerung bezieht sich überwiegend auf die Steuerung von Audio- oder Video-Medien (z. B. Play, Stopp, Pause...), die z. B. in Anlehnung an Videorecorder gestaltet werden sollte.

In der DIN EN ISO 14915 Teil 2 wird Navigation definiert als Bewegung des Benutzers in und zwischen Medienobjekten oder Darstellungssegmenten, um einen Gegenstand, ein bestimmtes Thema oder einen spezifischen Teil der Information zu finden.
Wichtig ist hier, wie im Abschnitt 4.8.1 auch erläutert, zwischen Inhaltsstruktur und Navigationsstruktur zu unterscheiden.
Die Inhaltsstruktur ordnet Themen und Medien, die Navigationsstruktur zeigt mögliche Bewegungen zwischen Teilen des Inhalts. Für beide Aspekte muss es ein Konzept geben und bei allen Änderungen, Entscheidungen muss immer auf eine Trennung der beiden Aspekte geachtet werden.

Ein Teil der Multimedia-Norm ist ausschließlich der Navigation und Steuerung von multimedialen Anwendungen gewidmet.
Grundansatz ist auch hier, dass zuerst die Inhalte gut (im Sinne der Benutzer) strukturiert sein müssen, bevor eine angemessene Navigation entworfen werden kann. Oder anders ausgedrückt: Durch eine chaotische, inkonsistente Struktur wird es keine gute Navigation geben können.

Navigationsdesign
Navigationshilfen sollten immer vorhanden sein, wie eine Suchfunktion, Anzeige des Pfads, „wo man schon gewesen ist“.
Es werden verschiedene Arten von Navigationsstrukturen unterschieden, wie Navigation in linearen Strukturen, in Baumstrukturen und in Netzen. Zur Navigation in Baumstrukturen finden sich z. B. folgende Empfehlungen:

Wenn es für die Aufgabe geeignet ist, soll der Nutzer imstande sein:

- innerhalb einer Ebene in der Struktur vor und zurückzugehen,
- die Struktur aufwärts und abwärts zu navigieren,
- in größeren Schritten vorzurücken,

Inhaltsstruktur	Navigationsstruktur
Themen und Medien	Medienobjekte, Darstellungssegmente, Inhaltsblöcke
...nach Aufgaben nach Nutzungsfaktoren (z. B. Häufigkeit) nach Benutzer-Objekt/Informationsmodell (z. B. logische Gruppen)	...je nach „Bedarf“ linear hierarchisch (Baumstruktur) vernetzt in Kombination
Inhaltsblöcke müssen passenden Darstellungssegmenten[12] zugeordnet werden	Steuerungselemente, häufig analog zu physischen Steuerungen (z. B. Knöpfe eines Videorecorders)

[11] DIN EN ISO 14915 Teil 2 Multimedia-Navigation und Steuerung
[12] physische Umsetzung der Inhaltsstruktur

- zum ersten bzw. Haupt-„Layer“ zu gehen,
- zum Beginn oder Ende der Struktur zu gehen,
- zu einem Inhaltsverzeichnis, Index oder einer Suchfunktionalität zu navigieren, um zu anderen gewünschten Orten zu navigieren.

Desweiteren finden sich Empfehlungen für die Gestaltung von Links (elektronischen Querverweisen) oder für Navigationsfunktionen wie z. B. „Vorwärts“, „Rückwärts“ oder die „Suche“.

Die Aussagen zu Navigationstechniken und -strukturen lassen sich z. B. auch auf Software mit Browser-Oberfläche anwenden.

In Anlehnung an die Regeln der ISO 9241 sollten die Steuerungselemente von Multimedia-Anwendungen z. B. deutlich als Steuerungselemente erkennbar sein und logisch gruppiert sein. Interessierte Leser seien auf die Originalnorm verwiesen.

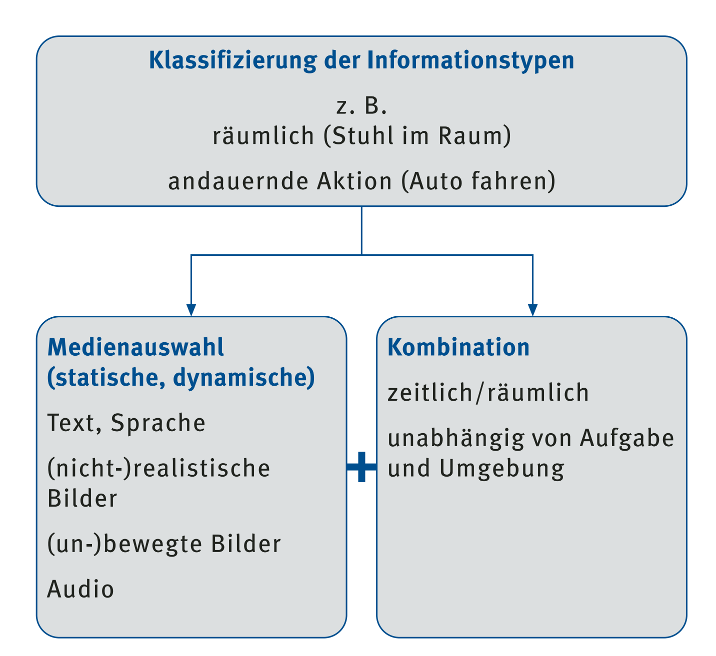
5.3.5 Auswahl und Kombination von Medien

Die im Teil 3 der ISO 14915[13] aufgeführten Richtlinien für den Einsatz mehrerer Medien beschreiben die Medienauswahl und -kombination in Abhängigkeit von den Kommunikationszielen und Aufgaben, für die sie eingesetzt werden. Medientypen sind z. B. Fotografien, Filme, Audio-Sequenzen.

Neben der Kategorisierung von Medien findet man dort eine Klassifizierung von Informationstypen. Informationstypen sind z. B. statische Informationen (Musik spielt), anlassbezogene Informationen (ein Telefon klingelt), Werte (quantitative Information) usw.
In der Norm werden insgesamt zwölf Informationsarten unterschieden und Beispiele dazu benannt.

Grundsätze zur Auswahl und Kombination von Medien sind:

- Auswahl der Medien nach Benutzereigenschaften (für ältere Benutzer keine schnellen Newsticker laufen lassen)
- Verwendung von Redundanz (Doppelungen) bei kritischen Informationen
- Vermeidung widersprüchlicher Wahrnehmungskanäle
- Vermeidung semantischer Widersprüche (wie Sie rechts von mir sehen...)
- Auswahl von Medienkombinationen zur detaillierten Darstellung von Informationen (ein reales Video eines Produktionsablaufs und eine grafische Animation für Details)
- Anwendung statischer Medien für wichtige Nachrichten



Auswahl und Kombination von Medien in Abhängigkeit von den Kommunikationszielen und Aufgaben ISO 14915 Teil 3

[13] ISO 14915 Teil 3 Auswahl und Kombination von Medien



Beispiel:
Video für Notfälle in Flugzeugen

- ***Verhaltensinstruktionen in sprachlicher Form darstellen***
- ***Notfallwege in grafischer Form darstellen***



Beispiel:
Firmenpräsentation

- ***einen Produktionsablauf als Video zeigen***
- ***den Umsatz des Unternehmens in grafischer Form darstellen***

Beispiele für Medienkombinationen

Bei der Auswahl und Kombination von Medien sollte man im ersten Schritt die Art der Information bestimmen und im zweiten Schritt die passenden Medien auswählt: Zum Beispiel als Vorbereitung auf eine Notfallsituation im Flugzeug wird den Passagieren ein Video gezeigt, die Verhaltensinstruktionen werden über Sprache (des Flugbegleitpersonals) vermittelt und zum Einprägen der Notfallwege wird zusätzlich eine Grafik gezeigt.
In der Norm finden sich zahlreiche Beispiele für die Bestimmung der Informationsart und die Auswahl und Kombination der passenden Medien.
Darüber hinaus gibt es Hinweise zur Medienintegration:

- Vorspann
- synchronisierte, in Beziehung stehende Medien
- Trennen von Quellen mit Audio-Inhalt
- Vermeidung von Störungen in Audio-Medien
- Begrenzung von Sprachunterbrechungen in Audio- oder Sprachmedien
- Integration von nichtrealistischen Bildern in realistische Bilder
- Benutzung von Bildunterschriften

Eine gute e-learning-Software (computer based training) berücksichtigt neben den Prinzipien der Ästhetik, der Wahr-

5.3.6 Leitsätze für e-learning-Software

nehmungs- und Kognitionspsychologie und der in dieser Broschüre behandelten Software-Ergonomie auch ein didaktisches Konzept.
Hier soll keine Argumentation für oder gegen den Einsatz von e-learning-Software erfolgen. Aber wenn die oft genannten Vorteile des e-learning greifen sollen, ist es offensichtlich, dass ergonomische Kriterien erfüllt sein müssen, wie:

- Die Lerninhalte sind schnell zugänglich und immer aktuell.
- Die Lerngeschwindigkeit lässt sich individuell anpassen.
- Das Lernen mit der Software wird als motivierend und inspirierend empfunden.

Anforderungen an die Software sind z. B.

- geführte, selbsterklärende Installation (bei CDs)
- Systemvoraussetzungen werden spezifiziert bzw. automatisch geprüft
- übersichtlicher und logischer Aufbau der Seiten
- Ausstieg und Wiedereinstieg jederzeit möglich
- optisch ansprechende Gestaltung
- keine zu große Textmenge pro Seite
- Exploration möglich (entdeckendes Lernen)
- Lernzielkontrolle am Ende der Module
- Feedback in Form von Bestätigungen, Korrekturen und Hinweisen
- Umsetzung und Anwendungsmöglichkeiten sind klar
- Suchfunktionen, Index und Glossar vorhanden

Zusammenfassend kann man die folgenden Faustegeln nennen:

5.4 Benutzerzentriertes Media-Engineering

Neben den im vorherigen Abschnitt beschriebenen Produktkriterien für Multimedia spielt auch hier, wie im Usability-Bereich allgemein, der Prozess, d.h. die Art und Weise, wie das Produkt erstellt wird, eine entscheidende Rolle.
In der Norm DIN EN ISO 14915 ist ein sogenannter 3-Stufen-Rahmen für Multimedia-Design (Content-Interaction-Media-Design) beschrieben, d. h. die Anforderungen an den Entwicklungsprozess finden sich hier nicht in einer gesonderten Norm, wie bei der ISO 9241 und der ISO 13407, sondern als integraler Bestandteil. Siehe am Ende Abschnitt 6.1.

Auch in der Medienproduktion gelten „Murphys Gesetze“! Umso wichtiger ist es, in diesem kreativen Bereich strukturierte Methoden aus dem Software-Engineering einzusetzen. Nicht allein Layout und Grafik, Schnitt und Timing entscheiden über den Erfolg eines Multimediaprodukts, auch nicht das Marketing, sondern vor allem das perfekte Zusammenspiel von Inhalt und Form.

Schritte bei Multimedia-Produktionen sind

- Projektplanung (Stufen der Softwareentwicklung berücksichtigen) und
- Kooperation verschiedenster Spezialisten.
- Obwohl die Usability-Kriterien unabhängig von der verwendeten Technologie zu beurteilen sind, ist es dennoch entscheidend, welche Technologien eingesetzt werden (Programmiersprache, Datenbank, Autorensystem...). Von der eingesetzten Technologie hängen viele Gestaltungsmöglichkeiten ab, wie Layout, Suchfunktion usw.
  Auch die eingesetzten Autorensysteme zur Entwicklung multimedialer Lehr-/Lernsysteme ermöglichen oder begrenzen die Gestaltung.

Vom Prozess her kann man einerseits vorgehen wie bei anderen Software-Produkten auch, d. h.

- Nutzungskontext erheben, Zielgruppen bestimmen, User-Szenarien recherchieren, User-Szenarien konsolidieren, Anforderungen ableiten, Prototyp validieren usw.

Hinzu kommen aber Schritte wie

- Medienpsychologische Konzeption,
- ggfs. didaktische Konzeption und Storybord,
- Mediendesign/Auswahl,
- Medienrealisierung,
- Medienschnitt.



Quelle: K. Wunderlich: Media Engineering Veranstaltung auf der informatica feminale 2000

Beim Durchlaufen dieser Phasen werden je nach dem, wie es passt, bestimmte Methoden eingesetzt. Die Methoden, um Multimediasysteme benutzungsfreundlich zu gestalten, sind im wesentlichen dieselben, wie bei anderen Softwareentwicklungen. Eine andere Methode, die speziell für diesen Bereich geeignet ist, ist das Arbeiten mit Storyboards.
Die Arbeit mit Storyboards ist eine Technik, die aus der Film- und Multimediaproduktion stammt. Mit Hilfe von Storyboards werden die visuellen Elemente und der didaktische Aufbau der Site geplant. Storyboards können helfen, den Inhalt und Aufbau jeder Seite zu planen und die Navigationselemente und -wege festzulegen, die ein Besucher auf einer Website wählen kann.
Das Storyboard ist das „Drehbuch" der Anwendung. Die geplanten Inhalte der Anwendung werden nach logischen Zusammenhängen und Prioritäten gegliedert. Das kann in einem einfachen System nur eine logische systematische Gliederung sein. In einem komplexeren System beinhaltet das Storyboard zusätzlich Anweisungen für die Programmierung.

5.5 Zusammenfassung

- Medieneinsatz nur zweckgebunden
- Nutzungskontext berücksichtigen
- Redundanz bei kritischen Informationen
- Kognitive Mehrbelastung vermeiden
  - Vermeiden widersprüchlicher Wahrnehmungskanäle
  - Vermeiden von Überlastung aufgrund zeitabhängiger Informationspräsentation
  - Vermeiden von Überlastung durch zusätzliche Aktivitäten
- Medienkombination für unterschiedliche Gesichtspunkte
- angemessene Reaktionszeiten
- gute Synchronisation
- Berücksichtigung von Wahrnehmungsunterschieden[14]
- Auswahlmöglichkeit zwischen visuellen und akustischen Erläuterungen

[14] Hiermit ist gemeint, dass ggf. Nutzergruppen mit eingeschränkten Fähigkeiten berücksichtigt werden sollten.



6 Barrierefreie Software-Gestaltung

6.1 Gestaltung barrierefreier Software

Der Grundgedanke barrierefreier Gestaltung ist:

- die Benachteiligung von behinderten Menschen zu beseitigen und zu verhindern,
- die gleichberechtigte Teilhabe von Behinderten am Leben in der Gesellschaft zu gewährleisten und
- ihnen eine selbstbestimmte Lebensführung zu ermöglichen.

Bezogen auf Software heißt dies, dass ihnen alle akustischen und visuellen Informationsquellen und Kommunikationseinrichtungen zur Verfügung stehen müssen, so dass sie sich z. B. bei öffentlichen Stellen, der Jobsuche oder beim Internet-Banking usw. genauso informieren können wie andere Menschen.
Barrierefreie Gestaltung (Accessibility = Zugänglichkeit) heißt, das die Software für behinderte Menschen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sein muss.
Behindert sind Personen, „wenn ihre körperliche Funktion, geistige Fähigkeit oder seelische Gesundheit mit hoher Wahrscheinlichkeit länger als sechs Monate von dem für das Lebensalter typischen Zustand abweichen und daher ihre Teilhabe am Leben in der Gesellschaft beeinträchtigt ist.“[1]
Mit der Veröffentlichung der Zugänglichkeitsrichtlinien des internationalen W3C[2] Gremiums im Mai 1999 und dem Inkrafttreten des Behindertengleichstellungsgesetzes (BGG) in Deutschland im Mai 2002 bestehen sowohl Anforderungen als auch zeitliche Fristen für deren Umsetzung.

Als Faustregeln barrierefreier Software gelten:

- klare Menüstruktur, einfache Navigation
- alle Informationen stehen als Text zur Verfügung (nicht nur als Grafiken oder Symbole)
- bei Online-Formularen die Felder für Einträge vorbelegen
- Grafiken und Bilder mit „Alt-Text“[3]
- Trennung von Information und Layout
- Website auch ohne Java und Flash[4] verständlich
- Schriftgröße kann vom Benutzer variiert werden
- klare Farbkontraste
- einfache Textgestaltung (Zusammenfassung, Inhalt, weiterführende Informationen)
- eindeutige Bezeichnung der Links (nicht „hier Klicken“)
- Abschriften von gesprochenen Audioclips verfügbar machen, Videos mit Untertiteln versehen



Quelle: Picture Alliance

Bei Sehbehinderungen ist es wichtig, dass die Schriftgröße vom Benutzer variiert werden kann. Dies wird durch barrierefreie Software ermöglicht.

[1] Behindertengleichstellungsgesetz

[2] World Wide Web Consortium , siehe Abschnitt 5 dieses Kapitels

[3] „Alt-Text“ bedeutet, das anstatt bzw. zusätzlich zum Bild ein kurzer aussagekräftiger Text das Bild bzw. die Grafik beschreibt.

[4] Java ist eine Programmiersprache. Kleine Programme, sogenannte Applets, die in Java programmiert sind, können Aussehen und Inhalte einer Website verändern. Flash ist ein Computerprogramm für die Erstellung von Animationen auf Webseiten.

6.2 Gründe für barrierefreie Software

Es gibt mehrere Gründe, warum auch nicht Betroffene oder Verantwortliche dieses Kapitel lesen sollten. Die Berücksichtigung der Regeln für barrierefreie Softwaregestaltung macht allen Nutzern das Leben leichter.

1. Abgesenkte Bordsteinkanten, sich automatisch öffnende Türen sind nur zwei Beispiele, mit denen deutlich wird, dass barrierefreie Gestaltung auch für Menschen ohne Beeinträchtigungen einen Nutzen hat. Auf Software übertragen ist eine klare Navigationsstruktur auf Webseiten für jeden Nutzer erfreulich. Eine klare Navigationsstruktur erleichtert akustisches Browsen für Sehbehinderte. Browser für Nicht-Sehende springen von Überschrift zu Überschrift, von Hyperlink zu Hyperlink und lesen den Text über eine Sprachausgabesoftware vor.
   Ein weiteres Beispiel ist die Maussteuerung, die Menschen mit motorischen Schwächen Probleme bereitet. Ermöglicht man die Navigation innerhalb eines Formulars auch mit der Tab-Taste in erwartungskonformer Weise, ist damit nicht nur den Menschen mit motorischen Schwächen gedient. Denn man kann davon ausgehen, dass Anfänger überwiegend mit der Maus arbeiten, geübte Benutzer aber gerne mit den Fingern auf der Tastatur bleiben und dort ebenfalls mit der Tab-Taste arbeiten.

2. Behindertengerechtes Design führt zu einer technisch zeitgemäßeren Architektur der Software, was wiederum die Flexibilität von Webseiten verbessert. Eine saubere Trennung von Inhalt, Struktur und Formatierung wird immer zwingender. Dies gilt auch für die Optimierung der Darstellung auf unterschiedliche Ausgabeformate. Gleicher Inhalt muss für verschiedene Ausgabegeräte (verschiedene Bildschirmauflösungen, PDAs, Navigationssysteme usw.) aufbereitet werden.
   Auch die Wartbarkeit einer Website wird durch die korrekte Verwendung der Technologien verbessert. Assistive Technologien[5] interpretieren z. B. HTML und setzen den Inhalt für Benutzer um. Wenn Tabellendarstellungen falsch codiert werden, sind diese für die Benutzer nicht korrekt interpretierbar.

Braille-Zeile: Mit den Fingern wird unterhalb der Tastatur jeweils eine Textzeile abgetastet. Auf Knopfdruck schaltet man Zeile für Zeile weiter. Dicht oberhalb eines jeden Braillebuchstabens befindet sich eine Taste, mit der der Cursor positioniert oder ein Link angeklickt werden kann.

Quelle: Picture Alliance

[5] Spezielle Hardware oder Software, die es behinderten Menschen ermöglicht, ihre Aktivitäten auszuführen (Hardware, wie Braille-Zeile, Blickverfolgungssysteme oder Software, wie Bildschirmlupe oder Screenreader)

3. Es gibt Bezüge zum sensiorengerechten Design. So wird gefordert, dass zeitgesteuerte Änderungen des Inhalts einer Website (z. B. Pop-Up-Fenster, Öffnen neuer Fenster, Laufschriften) durch Nutzer kontrollierbar sein müssen.

4. Es gibt Beeinträchtigungen, von denen der Einzelne vielleicht gar nichts weiß, z. B. Rechtschreibschwäche oder Farbfehlsichtigkeit. In der Umgangssprache wird meistens von Farbenblindheit gesprochen. Völlige Farbblindheit (Achromatopsie) ist ein erblich bedingter Defekt, der relativ selten (1 von 50.000) vorkommt. Es handelt sich dabei um Störungen bei der Empfindlichkeit der Zäpfchen bzw. völliges Fehlen der Zäpfchen im Auge. Dies kann allerdings auch aus einer Krankheit oder einem Unfall (zerebrale Farbenblindheit) resultieren. Sehr verbreitet ist hingegen die Farbfehlsichtigkeit. So sind ca. 8 % der Männer aber nur ca. 0,4 % der Frauen farbschwach. Diese Menschen haben oft eine Blau-Gelb- oder Rot-Grün-Schwäche. Feststellen kann man dies mit pseudoisochromatischen Tafeln.

Will man auf der sicheren Seite sein, entwirft man Benutzungsoberflächen zunächst monochrom. Der ebenfalls verbreiteten Rechtschreibschwäche trägt z. B. die Suchmaschine „Google“ bereits Rechnung. Dort ist eine Software integriert, die gegenüber falsch geschriebenen Eingaben robust ist bzw. „nachfragt“, was man meinte und dabei den korrekt geschriebenen Begriff zeigt.

6.3 Die Anforderungen allgemein[6]

Die Zugänglichkeitsrichtlinien für Web-Inhalte des W3C umfassen die folgenden Kernforderungen:

1. Stellen Sie äquivalente Alternativen für Audio- und visuellen Inhalt bereit.
2. Verlassen Sie sich nicht auf Farbe allein.
3. Verwenden Sie Markup und Stylesheets und tun Sie dies auf korrekte Weise.
4. Verdeutlichen Sie die Verwendung natürlicher Sprache.
5. Erstellen Sie Tabellen, die geschmeidig transformieren.
6. Sorgen Sie dafür, dass Seiten, die neue Technologien verwenden, geschmeidig transformieren.
7. Sorgen Sie für eine Kontrolle des Benutzers über zeitgesteuerte Änderungen des Inhalts.
8. Sorgen Sie für direkte Zugänglichkeit eingebetteter Benutzerschnittstellen.
9. Wählen Sie ein geräteunabhängiges Design.
10. Verwenden Sie Interim-Lösungen.
11. Verwenden Sie W3C-Technologien und -Richtlinien.
12. Stellen Sie Informationen zum Kontext und zur Orientierung bereit.
13. Stellen Sie klare Navigationsmechanismen bereit.
14. Sorgen Sie dafür, dass Dokumente klar und einfach gehalten sind.

Nachzulesen unter www.w3c.de/Trans/WAI/webinhalt.html

[6] Hier wird nicht auf technische Details eingegangen. Nähere Infos finden sich bei den entsprechenden Quellen.

6 Barrierefreie Software-Gestaltung (Accessibility)

Gegliedert nach Gestaltungselementen kann man die Tipps wie folgt zusammenfassen[7]:

1. **Bilder und Animationen**
   Verwenden Sie das ‚alt‘-Attribut, um die Funktion des visuell Dargestellten zu beschreiben.

2. **Image maps**
   Verwenden Sie die Client-seitige ‚map‘ und Text für Hotspots.

3. **Multimedia**
   Stellen Sie für Audio-Material Untertitel und Abschriften sowie Beschreibungen von Video-Sequenzen zur Verfügung.

4. **Hypertext Links**
   Verwenden Sie Text, der auch ohne den Kontext sinnvoll ist; vermeiden Sie zum Beispiel: „hier Klicken“

5. **Struktur der Seiten**
   Nutzen Sie Überschriften, Listen und eine in sich schlüssige Struktur. Verwenden Sie soweit möglich CSS für Layout und Stil.

6. **Grafiken und Diagramme**
   Fassen Sie die Darstellung kurz zusammen oder verwenden Sie das ‘longdesc’-Attribut.

7. **Scripts, Applets und Plug-Ins**
   Stellen Sie alternative Inhalte für die Fälle bereit, in denen aktive Merkmale nicht barrierefrei sind oder nicht unterstützt werden.

8. **Frames**
   Beschriften Sie Frames mit dem ‘title’- oder ‘name’-Attribut. Verwenden Sie das ‘noframes’-Attribut und sinnvolle Titel.

9. **Tabellen**
   Ermöglichen Sie zeilenweises Lesen. Fassen Sie zusammen.

10. **Prüfen Sie Ihre Arbeit**
    Kontrollieren Sie den HTML-Code. Nutzen Sie Bewertungs-Werkzeuge und Browser, die nur Text darstellen, um die behindertengerechte Darstellung zu überprüfen. Verwenden Sie die Checkliste und Richtlinien des W3C.



Die normale Version einer Website (www.diepresse.com)



Die Textversion derselben Website

[7] Quick Tips der WAI (Web Accessibility Initiative)

6.4 Die Anforderungen nach Benutzergruppen

Die W3C[8]-Anforderungen umfassen insgesamt 66 Richtlinien zur Gestaltung barrierefreier Systeme. Dabei handelt es sich um 17 Muss-, 33 Sollte- und 16 Kann-Anforderungen. Auch die Anforderungen aus der BITV oder ISO-Norm sind in unterschiedliche Prioritäten gegliedert. Für genauere Informationen hierzu sei an Experten bzw. die entsprechenden Quellen im Internet verwiesen.

Das W3C-Gremium geht von fünf Kriterien zur Gestaltung barrierefreier Systeme aus: Barrierefreie Software ist wahrnehmbar, bedienbar, bietet gute Orientierung und Navigierbarkeit, ist verständlich und (Fehler-)robust.
Allgemein werden die Kriterien wie folgt beschrieben:

Wahrnehmbarkeit
Inhalte und Navigationselemente müssen so präsentiert werden, dass sie von jedem Nutzer wahrgenommen werden können (Ausnahme: Elemente, die nicht in Worten ausgedrückt werden können).

Bedienbarkeit
Alle Interaktionselemente sollen ohne Einschränkungen von jedem Nutzer bedienbar sein.

Navigierbarkeit/Orientierung
Benutzer erhalten ständig Unterstützung für die Orientierung und Navigation (Sitemap, breadcrumb).

Verstehbarkeit/Verständlichkeit
Inhalt und die Interaktionselemente sind so einfach und verständlich wie möglich zu gestalten.

Robustheit
(Web-)Technologien verwenden, die bestmögliches Zusammenwirken mit heutigen und künftigen assistiven Technologien bieten.

Diese fünf Anforderungen müssen für fünf Behinderungsarten konkretisiert werden. Nämlich für Menschen mit Hör-, Sprach- und Sehbeeinträchtigungen, Menschen mit motorischen Schwächen und Menschen mit Lern- und Leseschwäche. Das führt zu unterschiedlichen, teilweise sich widersprechenden Anforderungen. Im Einzelfall muss hier geklärt werden, welche Zielgruppe Priorität hat. Wie man der Übersicht auf der nächsten Seite entnehmen kann, kann das, was für die eine Benutzergruppe vorteilhaft ist, für eine andere nicht empfehlenswert sein.

[8] siehe Fußnote 2 auf S. 95

Anforderung/Behinderungsart	Wahrnehmbarkeit	Bedienbarkeit
Lese-, Lernschwäche	kurze Texte, die das Wesentliche in einfachen Worten wiedergeben / langsame Wiedergabeanimationen bei Text / starke Farbkontraste erhöhen Aufmerksamkeit / Texte mit Multimedia-Elementen ergänzen	interaktive Applets, Quick Time, Flash erregen eine höhere Aufmerksamkeit / animierte Farbkombinationen in der Interaktivität vermeiden, da epileptische Anfälle ausgelöst werden können
Hörbehinderung	kurze Texte, die das Wesentliche in einfachen Worten wiedergeben / starke Farbkontraste erhöhen Aufmerksamkeit / komplexe Texte mit erläuternden grafischen Elementen ergänzen / Untertitel für Video und Animation	
motorische Schwächen	interaktive Elemente als solche kenntlich machen / keine beweglichen interaktiven Links oder Buttons / korrekte Tab-Reihenfolge der interaktiven Elemente (von oben links nach unten rechts)	
Sprachbehinderung	kurze Texte, die das Wesentliche in einfachen Worten wiedergeben / komplexe Texte mit erläuternden grafischen Elementen ergänzen	
Sehbehinderte, Farbfehlsichtige		strukturierte HTML-Dokumente /Alternativen zu interaktiven Java-Applets[9] / Navigationsframes mit Bezeichnern kenntlich machen / Farbkombinationen für Interaktivität vermeiden / skalierbare Schriften in den interaktiven Elementen / Tab-Reihenfolge / Clientseitige Image maps

Es ist also jeweils vorab zu klären, in welchem Nutzungskontext und für welche Nutzergruppe die jeweilige Software genutzt werden soll. Dieses grundsätzliche Vorgehen ist bereits im „Nutzerzentriertem Entwicklungsprozess“ im Abschnitt 8.2 beschrieben.

Selbst wenn man es nicht schafft, alle Anforderungen sofort umzusetzen, ist doch schon mit der Umsetzung einiger Anforderungen etwas erreicht. Denn sicher ist es nicht einfach, eine Seite für alle Anwender benutzergerecht zu gestalten, es ist jedoch ein großer Unterschied zwischen nicht perfektem Design und völlig rücksichtslosem Design.

[9] Programme, die interaktive Teile in Dokumenten im Internet realisieren

6.5 Institutionen und Verordnungen

Grundlagen für die hier beschriebenen Anforderungen sind Forderungen des W3C und des BGG. Konkrete Formulierungen finden sich in der ISO 16071 und der Barrierefreien Informationstechnik-Verordnung – BITV.
Zu den genannten Institutionen bzw. Verordnungen finden sich hier in alphabetischer Reihenfolge kurze Erläuterungen.

Barrierefreie Informationstechnik-Verordnung (BITV)
Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz.
Diese Verordnung ist dem Behindertengleichstellungsgesetz nachgeschaltet und gilt für Behörden der Bundesverwaltung. Genauer deren Internet- und Intranet-Anwendungen und weiteren Angeboten, sofern sie öffentlich zugänglich sind (z. B. CD's).
Dem Standard liegen die Zugangsrichtlinien W3C zugrunde. In der BITV werden 14 Anforderungen formuliert, die das jeweils zu erreichende Ziel beschreiben. Diese Anforderungen werden durch Bedingungen in zwei Prioritätsstufen technisch konkretisiert. Priorität I soll unüberwindbare und signifikante Barrieren vermeiden, während die zusätzliche Berücksichtigung der Priorität II weitere Barrieren vermeidet und die Benutzung erleichtert. Seiten, die neu gestaltet oder in wesentlichen Bestandteilen verändert werden, müssen ab sofort dem Standard entsprechen. Seiten, die sich speziell an behinderte Menschen richten, mussten bis Ende 2003 und alle anderen Seiten bis Ende 2005 dem Standard entsprechen. Jeweils mindestens ein Zugangspfad zum barrierefreien Angebot muss barrierefrei angelegt werden.

Behindertengleichstellungsgesetz (BGG)
Das Gesetz zur Gleichstellung behinderter Menschen ist seit Mai 2002 in Kraft und gilt für Dienststellen und sonstige Einrichtungen der Bundesverwaltung (alle öffentlichen Stellen). Hierin werden diese Stellen verpflichtet, barrierefreie Informationstechnik für Mitarbeiter und Bürger zur Verfügung zu stellen.

Erfasst werden mit diesem Gesetz

- bauliche und sonstige Anlagen, Verkehrsmittel,
- technische Gebrauchsgegenstände (insbesondere auch Konsumgüter),
- Systeme der Informationsverarbeitung,
- akustische und visuelle Informationsquellen und Kommunikationseinrichtungen sowie
- andere gestaltete Lebensbereiche.

Gefordert wird die Einbeziehung von Internetseiten und grafischen Programmoberflächen.

www.behindertenbeauftragter.de/gesetzgebung/behindertengleichstellungsgesetz

ISO/TS 16071 Richtlinien für Barrierefreiheit an Mensch-Rechner-Schnittstellen (u.a. = Internetseiten und grafische Programmoberflächen).
Dies ist eine sog. Technical Specification, die nur in englischer Sprache vorliegt.
Hier sind Konzepte zusammengefasst, die einen universellen Zugang zu Informations- und Kommunikationstechnologien, insbesondere für Benutzer mit spezifischen Bedürfnissen in den Mittelpunkt stellen.

Dies sind

- Benutzer mit permanenten und temporären physischen, sensorischen und kognitiven Beeinträchtigungen,
- Benutzer, die mit mobilen oder technologisch eingeschränkten Geräten, z. B. ohne Maus oder Keyboard, mit begrenzten Übertragungsraten, nicht aktuellen Browser-Versionen, kleinen Displays oder nur alphanumerischen Anzeigemöglichkeiten auf Informationen zugreifen möchten.
- Benutzer, die keine oder nur eingeschränkte Kenntnisse der Sprache haben, in der der Content in einem System bereitgestellt wird.

Hier finden sich

- 71 Gestaltungsanforderungen
- 3 Prioritäten: core, primary, secondary
- 2 Bereiche: operating system, application

World Wide Web Consortium (W3C)
Das World Wide Web Consortium (kurz W3C genannt) und seine Mitglieder beschäftigen sich mit der Schaffung, Weiterentwicklung von Standards oder mit der Entwicklung neuer Standards für das Internet. Dieses Gremium hat z. B. auch den Standard HTML verabschiedet. Ohne derartige Standards würde das Internet nicht so funktionieren, wie wir es heute kennen.
Dieses Gremium hat die Web Content Accessibility Guidelines (WCAG) veröffentlicht.
http://www.w3.org/TR/WCAG
Web Content Accessibility Guidelines 1.0, Mai 1999

In deutscher Sprache:
http://www.w3.org/Consortium/Offices/Germany/Trans/WAI/webinhalt.html
Zugänglichkeitsrichtlinien für Web-Inhalte 1.0



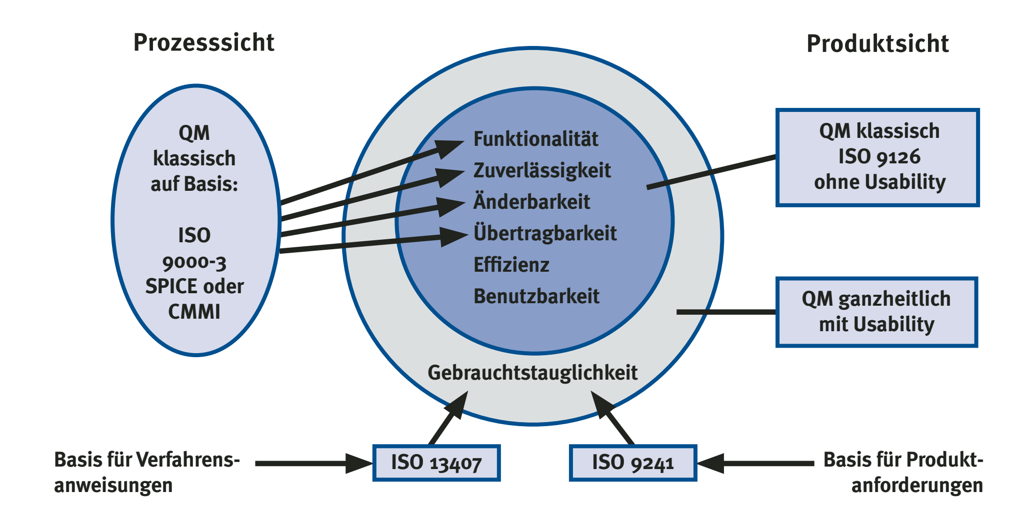
7 Ganzheitliche Software-Qualität

7.1 Von Ganzheitlichkeit und Qualität

Es ist billiger Qualität zu produzieren, als zu überprüfen.

Ebenso wie für viele Produkte gibt es auch für Software-Produkte Qualitätsmerkmale. In der Regel werden darunter softwaretechnische Merkmale wie Zuverlässigkeit, Funktionserfüllung, Verfügbarkeit, Sicherheit, Leistung, Wartungsfreundlichkeit (Änderbarkeit), Übertragbarkeit und Benutzbarkeit[1] verstanden.
Zuverlässigkeit wird z. B. definiert als Wahrscheinlichkeit, dass dieses Programm erfolgreich gemäß der Spezifikationen für einen gegebenen Zeitabschnitt ausführt wird.
Da Software-Produkte fortlaufend geändert werden, ist Wartungsfreundlichkeit ein wichtiges Kriterium, welches sich auf leichte Änderbarkeit und Lesbarkeit der Dokumentation bezieht und den Zeitaufwand für Wartungsvorgänge quantifiziert.
Interessant wird es beim Kriterium der Funktionserfüllung. Dies bedeutet, dass das Software-Produkt die in der Anforderungsspezifikation (Pflichtenheft) festgelegten Funktionen innerhalb vorgegebener Randbedingungen erfüllt. Sofern es sich hierbei um Festlegungen handelt, ist eine Prüfung relativ einfach. Ein großer Teil der Anforderungen wird jedoch in Umgangssprache formuliert.
Dieses führt oft zu Unvollständigkeit und Mehrdeutigkeit. Z. B. die Anforderung, der Arbeitsablauf soll unterbrechbar sein, muss in Hinblick auf die Programmierung genauer formuliert werden. Außerdem werden in Pflichtenheften[2] oft nur die einzelnen Funktionen beschrieben, ohne deren Anwendung während eines Arbeitsablaufs zu berücksichtigen. Um eine ganzheitliche Softwarequalität zu erreichen, ist es erforderlich, softwaretechnische um software-ergonomische Qualitätskriterien zu ergänzen. Software-ergonomische Kriterien sind Inhalt der Kapitel 3 und 4 dieser Broschüre.

Um eine hohe Nutzungsqualität einer Software zu erreichen, spielt neben der Qualität des Produktes auch die Qualität des Herstellungsprozesses der Software eine entscheidende Rolle. Diese Erfahrungen aus der industriellen Produktion lassen sich auf Software übertragen. Hierzu gibt es verschiedene Ansätze.
Ziel der ISO 9000 als Sammlung branchenunabhängiger, weltweit gültiger Normen zum Qualitätsmanagement ist die Zertifizierung von Qualitätsmanagement-Systemen, wie sie in den Unternehmen angewendet werden. Qualitätsmanagement zielt darauf ab, den Wert des Produktes für den Kunden und den Benutzer zu steigern.



Usability und Qualität

[1] Erläuterungen dazu im Abschnitt 1.4

[2] Zur vertraglichen Gestaltung siehe Abschnitt 9.1

An der alten Normfassung gab es viel Kritik, u.a. dass viel Papier produziert wurde, sich aber in der Praxis wenig verbessert hat, da es keine Forderung nach kontinuierlicher Prozessverbesserung gab. Die derzeit aktuelle Fassung ist die DIN EN ISO 9000:2000.

Als Teil Normenfamilie entstand die ISO 9000-3. Diese Norm befasst sich mit der Entwicklung, Lieferung und Wartung von Software, d. h. dort, wo Software auf Vertragsbasis entwickelt wird. In dieser Norm ist berücksichtigt, dass einzelne Qualitätsaktivitäten in speziellen Phasen des Entwicklungsprozesses stattfinden (z. B. Review zu den Abnahmekriterien), andere Aktivitäten während des gesamten Entwicklungsprozesses angewendet werden sollten (z. B. Qualitätsaufzeichnungen, wie Testberichte).
Ein weiterer Ansatz zum Software-Qualitätsmanagement kommt ursprünglich aus den USA. Nachdem das amerikanische Verteidigungsministerium bei seinen Software-Lieferanten fortwährende Überschreitung von Lieferterminen und Entwicklungsbudgets sowie mangelnde Qualität der Softwareprodukte feststellte, erteilte es 1986 den Auftrag für das sogenannte CMM-Modell [3](jetzt CMMI®).
Dieses Modell soll helfen, den Reifegrad von Softwareentwicklungsprozessen (von den Lieferanten) zu ermitteln, um diese beurteilen und um gezielt Verbesserungen vornehmen zu können. Die Idee ist, im Gegensatz zur ISO, dass man nicht „von 0 auf 100“ zertifiziert, sondern dass verschiedene Reifestufen des Prozesses erreichbar sind. Erst mit steigendem Reifegrad wird die Erwartung verbunden, dass die Vorhersagbarkeit von Terminen, Kosten- und Qualitätszielen zunimmt.
CMMI unterscheidet fünf unterschiedliche Reifegrade für die entwickelten Fähigkeiten einer IT-Organisation: Initial, Wiederholbar, Definiert, Gemanagt und Optimiert. Für jede der zuvor genannten Stufen definiert das CMMI eine Reihe von Prozessgebieten und Fähigkeiten, die eine Organisation in dieser Stufe durchführt. Das Zertifizierungsverfahren für benutzerzentrierte Entwicklungsprozesse (DIN EN ISO 13470) ist an diese Stufung angelehnt, umfasst aber nur drei Stufen.
In der niedrigsten Stufe ist der Usability-Engineer als Berater tätig, in der nächst höheren Stufe ist der Usability-Engineer Mitglied im Design-Team (disziplinierter Prozess) und in der höchsten Stufe koordiniert der Usability-Engineer alle Design-Entscheidungen (sich ständig verbessernder Prozess.)

Seit 1993 gibt es SPICE[4] ISO TR 15504. Dies ist ein international anerkanntes Normungsprojekt zur Bewertung und Verbesserung von Prozessen in Organisationen, die Software entwickeln. Das Projekt soll die Ansätze von ISO 9000 und des CMM weiterentwickeln. Dort werden Anforderungen an die einzelnen Teile des Software-Entwicklungsprozesses gestellt und deren sukzessive Verbesserung angestrebt.

Im Gegensatz zur ISO sind CMM und SPICE zwar spezifische Normen zur Softwareentwicklung, beinhalten aber keine spezifischen Methoden zur Integration von Benutzeranforderungen.

Diese Lücke schließt seit 1997 die DIN EN ISO 13407 „Benutzer-orientierte Gestaltung interaktiver Systeme“. Im Abschnitt 8.2 sind die einzelnen Prozessschritte genauer beschrieben. Danach sollen Usability-Aktivitäten in die Entwicklung von Anwendungen integriert werden.

Als Maßnahmen sind vorgeschlagen:

- Ausführung und Koordination von Maßnahmen zur Absicherung eines benutzerzentrierten Entwicklungsprozesses
- Mitarbeit in Entwicklungsteams
- Unterstützung des Entwicklungsteams beim GUI-Design
- Durchführung von Usability Reviews
- Software-Ergonomie-Training für das Entwicklungsteam
- Durchführung von Aufgabenanalysen
- Etablierung von Unternehmens-Standards für GUIs
- Entwicklung und Evaluation von Prototypen.

Angewendet werden soll die Norm bei Software-Herstellern. Zur Prüfung dieser Norm gibt es einen „Leitfaden für die Evaluierung des Usability-Engineering-Prozesses bei der Herstellung und Pflege von Produkten“[5]. Das dort beschriebene Prüfverfahren lehnt sich an das CMM-Modell an. Danach ist es möglich, dass ein Usability Engineer[6] lediglich als (unverbindlicher) Berater tätig ist (zufälliger Prozess), dass er Mitglied des Design-Teams ist (disziplinierter Prozess) oder sogar alle Design-Entscheidungen koordiniert (sich ständig verbessernder Prozess).

[3] Capability Maturity Model (Reifegrad-Modell) – Integration
[4] SPICE = Software Process Improvement (Verbesserung) and Capability dEtermination (Bewertung)
[5] Prüfbaustein Usability Engineering-Prozess http://www.datech.de
[6] Person, die durch entsprechende Qualifikation diese Rolle einnimmt. Hauptaufgabe ist das Management der Umsetzung der Nutzer-Anforderungen

7.2 Kostentransparenz im Software-Lebenszyklus

„Ein Promille weniger Fehlbedienung bei Kopierern spart 100 Millionen Blatt Papier pro Jahr.“ [7]

In vielen Fällen werden hinsichtlich der Nützlichkeit des Computereinsatzes Angaben zur Verarbeitungsgeschwindigkeit oder der Funktionsvielfalt gemacht. Die Nützlichkeit eines Softwareproduktes ergibt sich aber nicht nur aus technischen Merkmalen. Zu betrachten sind die Nutzungskosten. Dies sind Kosten für Benutzersupport, Wartung, Customizing, Systemverwaltung und Schulung. Diese Kosten sind zusammen oft wesentlich höher als die eigentlichen Anschaffungskosten der Software.
Ziel des systematischen Einsatzes von Methoden des Usability Engineerings ist eine Senkung der Nutzungskosten. Die Methoden sind im Abschnitt 8.3 beschrieben.
Laut einer Studie von IBM spart jeder in Usability Tests investierte Dollar bis zu 100 Dollar Supportkosten.

Eine Untersuchung des Usability Spezialisten Jacob Nielsen[8] hat die jährlichen Kosten für die Nutzung eines Intranets bei unterschiedlicher Usability berechnet. Diese betragen für Firmen mit 10.000 Intranet-Anwendern:

- bei hoher Usability (unter den besten 25 %): $ 15,6 Millionen
- bei durchschnittlicher Usability: $ 20,7 Millionen
- bei geringer Usability (unter den schlechtesten 25 %): $ 30,4 Millionen

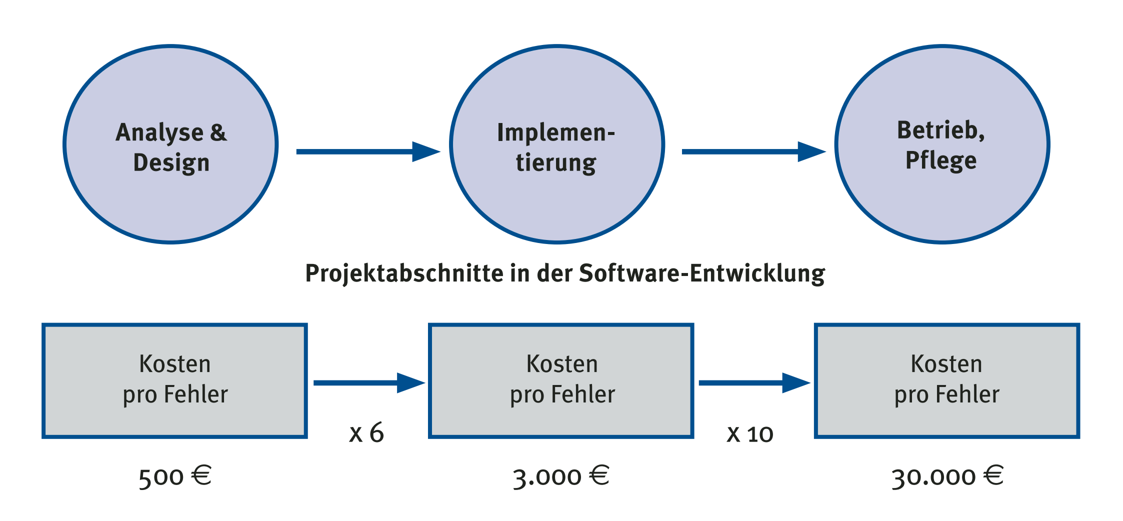
So können Mitarbeiter für die Erledigung ein und derselben Aufgaben (z. B. das Auffinden von Informationen über Kollegen oder das einer bestimmten Sozialleistung) via Intranet bis zum fünffachen Zeitaufwand benötigen, je nachdem, ob ihnen ein sehr benutzerfreundliches oder ein sehr kompliziert zu handhabendes Intranet zur Verfügung steht.

Interessant ist auch die 1991 von F. Brodbeck in einer Studie veröffentliche Zahl, dass 10 % der Arbeitszeit am Computer mit der Bewältigung von Fehlern verbracht werden.[9] Fehler machen Benutzer oft, wenn die Software nicht gut an ihre Arbeitsabläufe angepasst wurde.
Die Entwicklung der Benutzungsschnittstelle nimmt einen erheblichen Teil der Gesamtkosten von Software-Entwicklungsprojekten ein, einige Angaben liegen bei bis zu 60 %. Betrachtet man die Kostenverteilung während des gesamten Lebenszyklus eines Software-Produktes – also Entwicklungsphase + Nutzungsphase – gehen Schätzungen dahin, dass ca. 33 % auf die Entwicklung selbst entfallen, aber 67 % auf so genannte Wartungs- und Pflegekosten. Allgemein bekannt ist, dass es sich bei diesen Kosten oft eher um Weiterentwicklung als um „Wartung“ handelt.
Ganzheitlich betrachtet müsste man stets Entwicklungs- und Nutzungskosten zusammen als Maßstab für Rentabilitätsrechnungen nehmen. In der Praxis ist es jedoch oft so, dass es einen Verantwortlichen für das Projekt gibt, dessen Interesse darin liegt, zu möglichst geringen Kosten das Projekt abzuschließen. Für das Einbringen von softwareergonomischen Verbesserungen kann dies bedeuten, dass diese im Projekt abgelehnt werden, da die erhöhten Nutzungskosten erst nach Projektende entstehen.

Software-Entwicklungsprojekte laufen nicht selten aus dem vorgesehenen Zeit- und Kostenrahmen.

Immer wieder werden software-ergonomische Verbesserungswünsche aus Kostengründen abgelehnt.
Dies ist meistens dann der Fall, wenn Projekte beendet sind und sich nach einigen Wochen Benutzerbeschwerden häufen und die Benutzer mit so genannten „work arounds“ (drumherum arbeiten) das Problem umgehen. Eine quantitative Erfassung dieser work arounds macht Produktivitätsverluste deutlich. Eine sogenannte „Nachkalkulation“, wie sie in der industriellen Produktion üblich ist, findet für Software-Entwicklungen höchst selten statt. Häufig ist es so, dass die Kosten umso niedriger sind, je früher ein Fehler im Entwicklungsprozess entdeckt wird.

[7] Toshiba 1994
[8] Useit.com November 2002
[9] Zitiert aus: www.sozialnetz-hessen.de/ergo-online



Kostenfortpflanzung bei Fehlentscheidungen

Hauptursache für hohe unnötige Nutzungskosten ist, dass die Anforderungen an die Nutzung nicht ausreichend ermittelt wurden. Im Usability Engineering unterscheidet man deshalb zwischen Anforderungsermittlung und Anforderungsdefinition.
Die Ermittlung dessen, was die Benutzer wirklich brauchen, erfolgt mittels der Kernmethoden: Nutzergruppen- und Kontextanalyse und der Erhebung von Use Szenarien und der anschließenden Bestätigung durch Benutzer anhand von visualisierten Entwürfen (Prototypen). Erst wenn die Anforderungen ermittelt wurden, kann man sie formal definieren. Ein so praktiziertes Vorgehen führt zu mehrfachen Spareffekten[10] durch:

- Steigerung der Effektivität
- Verringerung der Benutzungsfehler um das Fünffache
- Einsparungen bei der Entwicklung
- Einsparungen bei der Schulung
- Geringere Wartungs- und Pflegekosten

Will man quantitative Argumente für software-ergonomische Verbesserungen sammeln, kann man folgende Werte zugrunde legen:

- Zahl der betroffenen Benutzer
- Zahl der betroffenen Benutzeraktionen (z. B. Zahl der Masken, Eingaben o. ä.)
- Zahl der vermiedenen Nutzungsfehler
- eingesparte Arbeitszeit (in Sekunden/Minuten)
- eingesparte Trainingskosten für Benutzer
- eingesparte Implementierungskosten (bis zu 6-fach)
- eingesparte Wartungskosten, Nutzungsdauer des Produkts

[10] Landauer, T.K. (1995). The Trouble with Computers. The MIT Press, Cambridge MA, London. ISBN 0-262-12186-7

Eine Berechnung kann man auch für Eigenentwicklungen aufstellen:

Annahme: Eine ergonomische Verbesserung trägt dazu bei, dass nur 1 Sekunde pro Maske eingespart wird:

Berechnungsgrundlage	Einsparung
250 Arbeitskräfte in einer Versicherung bearbeiten jeweils 27 Masken an 220 Arbeitstagen im Jahr. Setzt man 18 € Gehalt pro Stunde an, so ergibt sich bereits unter diesen extrem konservativen Annahmen eine Einsparung von	7.425 € pro Jahr (250 x 27 x 220 x 1/3600 x 18) Bei einer Zeiteinsparung von 10 Sekunden spart man bereits etwa 74.000 € Bei einer Minute kommen fast 450.000 € zusammen. (vielleicht mehr als der Anschaffungspreis der Software).

In dieser Kalkulation sind die durch ergonomische Gestaltung verringerten Nutzungsfehler und die geringeren Kosten für Schulung und Benutzerbetreuung nicht enthalten.

Abschließend sei noch auf folgendes hingewiesen:
In jedem Software-Entwicklungsprojekt gibt es mindestens Planungen/Kostenschätzungen für die Erstellung einer Online-Hilfe, von Benutzerhandbüchern, Schulungen u. ä. Hier sollte bereits zu Beginn darauf hingewiesen werden, dass Handbücher und Hilfefunktionen nachweislich nur von wenigen Benutzern und selten verwendet werden.

Nutzer bevorzugen bei Problemen den direkten und schnellen Kontakt zu anderen Kollegen oder „Key Usern“. Diese sind schnell erreichbar und vor allem brauchen diese sich nicht erst in das Problem hineindenken (wie möglicherweise) eine externe Hotline. Dieses sollte bedacht werden und es ist abzuwägen, ob ein Teil des Geldes, der für eine Hilfefunktion geplant war, nicht besser in einige Usability Engineering-Methoden investiert wird.



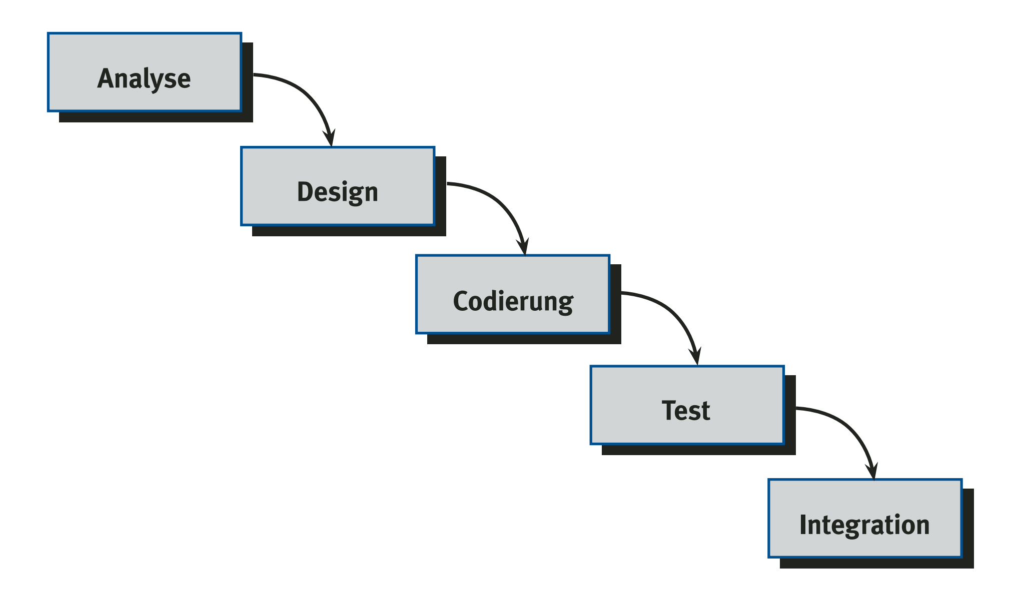
8 Der Software-Entwicklungsprozess

8.1 Vorgehensmodelle

Designing on requirements and walking on water is easy if both are frozen.

Jede Software-Entwicklung sollte in einem festgelegten organisatorischen Rahmen erfolgen. Der Prozess, also der zeitliche Ablauf der einzelnen Arbeitsschritte, sollte geplant, mit den Beteiligten abgestimmt und kontrolliert werden. Dafür wurden im Laufe der letzten Jahre verschiedene Vorgehensmodelle entwickelt, die abhängig von Branche und Projekt in zahlreichen angepassten Varianten und mit anderen Bezeichnungen Grundlage für Software-Entwicklungen sind.

Das **Wasserfallmodell** von Anfang 1970 hat seinen Namen daher, dass man davon ausging, eine Stufe/Phase nach der anderen zu bearbeiten und dass die Ergebnisse wie bei einem Wasserfall in die nächste Phase fallen. In der Praxis erwies sich dieses streng sequenzielle Vorgehen jedoch als unrealistisch. Benutzerbeteiligung war hier nur einmal am Anfang in der Definitionsphase vorgesehen.

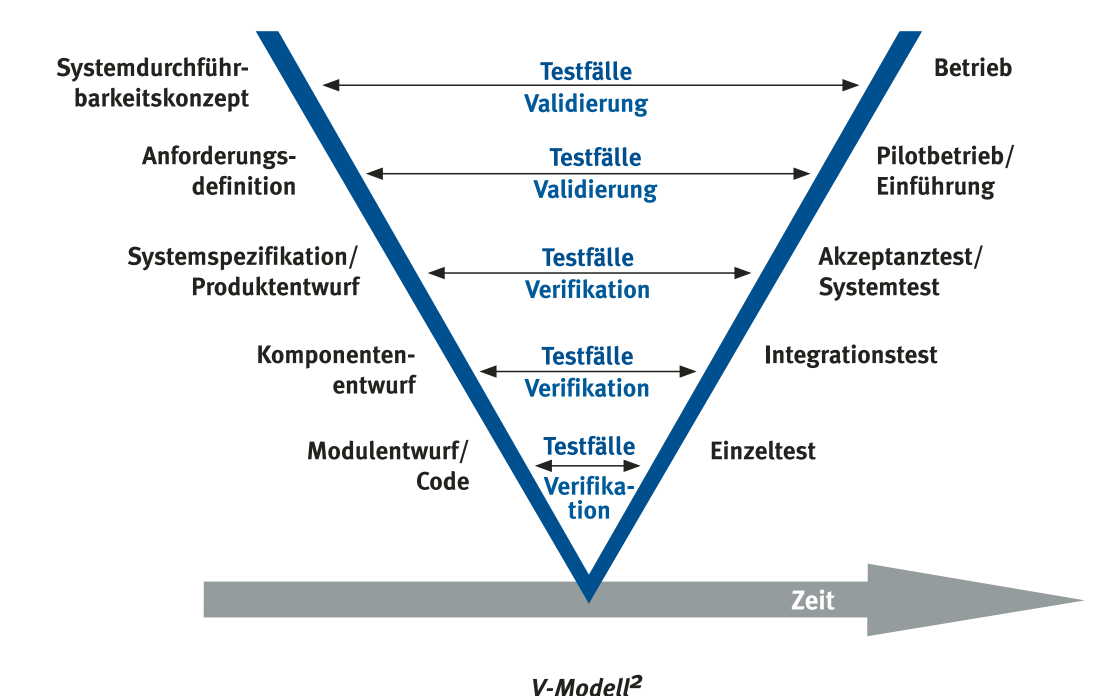


Wasserfallmodell

8 Der Software-Entwicklungsprozess

Das **V-Modell**[1]® („V“ steht für Vorgehen), das seit 2005 als V-Modell XT vorliegt, ist als Entwicklungsstandard für IT-Systeme des Bundes für die Planung und Durchführung von IT Projekten verbindlich vorgeschrieben.
Hier wird der Prozess in Aktivitäten (Tätigkeiten) und Produkte (Ergebnisse der Tätigkeiten) eingeteilt. Somit soll eine permanente Qualitätssicherung schon während der Entwicklung und nicht erst am Ende gewährleistet sein. Zwar wird eine Benutzerbeteiligung prinzipiell vorgesehen, aber es gibt, trotz entsprechender Bemühungen, zur Zeit noch viele offene Fragen hinsichtlich der konsequenten Einbettung ergonomischer Verfahrensschritte.

Positiv ist, dass die Validierung vor der Verifizierung erfolgt. Zur Erläuterung: In vielen Projekten wird lediglich durch eine Testphase verifiziert, d. h. geprüft, ob das Software-Produkt der Spezifikation (z. B. dem Pflichtenheft) entspricht.
Oft gibt es aber Missverständnisse beim Abgleich des Lastenheftes (fachliche Anforderungen) mit dem Pflichtenheft (technische Realisierung). Deshalb ist es sinnvoll, die fachlichen Anforderungen zu validieren, d. h. zu prüfen, ob sie gültig sind, wie sie genau zu verstehen, d. h. gemeint sind. Wird z. B. die Unterbrechbarkeit von Dialogabläufen gefordert, kann dies aus fachlicher Sicht anders gemeint sein, als es technisch zu realisieren ist.



V-Modell[2]

Validierung	Eignung bzw. der Wert des Produktes bezogen auf seinen Einsatzzweck	Wird ein passendes Produkt entwickelt? Am I building the right product?
Verifikation	Überprüfung der Übereinstimmung zwischen einem Produkt und seiner Spezifikation	Wird das Produkt richtig entsprechend den Vorgaben entwickelt? Am I building the product right?

[1] http://www.v-modell.iabg.de

[2] Die Abbildung entstammt der alten Version des V-Modells.

Beide Phasenmodelle haben allerdings den Nachteil, dass sie die folgenden in der Praxis vorkommenden Aspekte nicht berücksichtigen:

- Die Anforderungen an ein neues System sind zu Beginn nie vollständig bekannt.

- Während der Entwicklung kann sich eine produktive Zusammenarbeit zwischen Benutzern und Entwicklern ergeben, aus der neue Realisierungsmöglichkeiten resultieren.

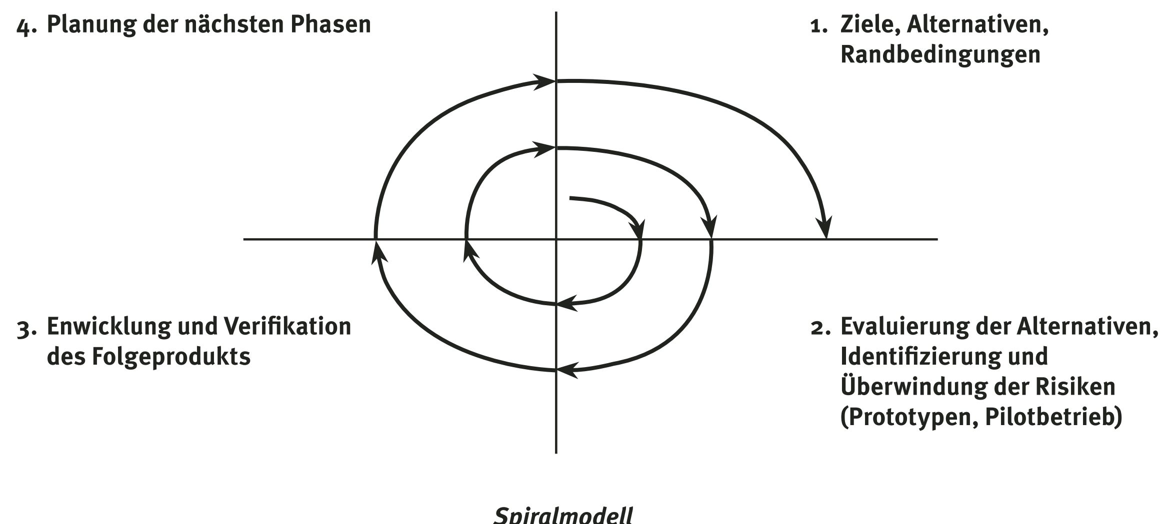
- Es gibt für eine Anforderung verschiedene Realisierungsmöglichkeiten.

Die Realisierung bestimmter Anforderungen lässt sich nicht theoretisch garantieren, d. h., ihre Realisierung muss vorgezogen werden.

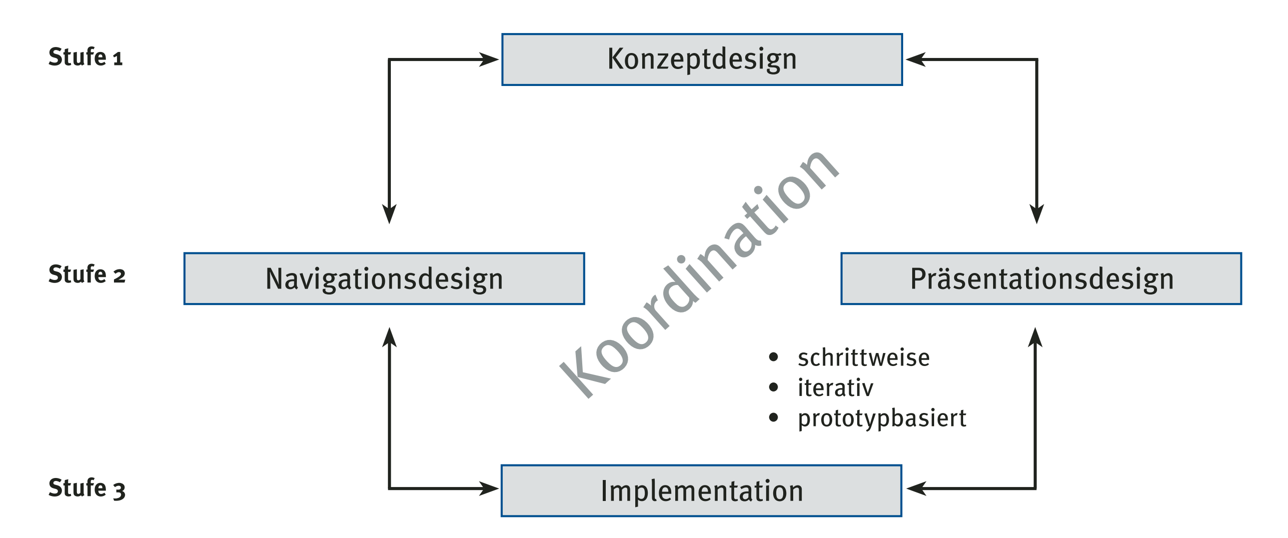
Das **Spiralmodell** gliedert sich in 4 Stufen, die für jede Verfeinerungsstufe der Software-Entwicklung bzw. jedes Modul (Teilprodukt) erneut durchlaufen werden. Nach der Identifikation der Ziele und Entwicklung alternativer Lösungsmöglichkeiten erfolgt die Bewertung der Lösungsmöglichkeiten. Abhängig von den Ergebnissen wird dann das Vorgehen für diesen Schritt festgelegt. Dieses kann in einem Fall z. B. mit Prototypen erfolgen, ein anderes Mal ohne Prototypen. Dieses Modell hat den Vorteil, dass die Ziele für den nächsten Zyklus, von den Ergebnissen des vorherigen abhängen und man je nach Komplexität der Teilaufgaben flexibel vorgehen kann.

Interessant ist auch, dass hierbei keine Trennung zwischen Entwicklung und Wartung gemacht wird. Das heißt, man geht davon aus, dass an einer Software ohnehin ständig weitergearbeitet wird und diese nicht am Tag x „fertig“ ist.

Was sich für den Bereich der grafischen Oberflächen erst entwickeln musste, wurde mit dem WWW zur Selbstverständlichkeit. Da gute Web-Seiten (gut heißt, z. B. für ein Versandhaus, dass ausreichend Umsatz gemacht wird oder für ein Intranet, dass es von vielen Beschäftigten genutzt wird) einem ständigen Verbesserungsprozess unterliegen, ist hier das iterative Vorgehen üblich. Ist der Relaunch (engl. Neu-Start) einer Web-Site erfolgt, wird bereits mit der Planung des nächsten Relaunch begonnen. So erfolgt quasi ein ständiges Management der verschiedenen Anforderungen auf das demnächst online gehende Software-Produkt.



Spiralmodell



Web-Engineering-Modell
3-Stufen-Rahmen für Multimedia-Design
ISO 14915 Content-Interaktion-Media-Design

Interessant an dem abgebildeten **Web-Engineering-Modell** ist die Reihenfolge der Schritte. Die so genannte Content-Konzeption (das heißt, was soll inhaltlich (Content) alles auf der Web-Site zu finden sein) steht an erster Stelle, danach folgt die Konzeption der Navigation und das Layout der Seiten. Diese beiden Aktivitäten werden dabei immer wieder zu gemeinsamen Ergebnissen zusammengeführt, erst danach erfolgt die Programmierung.

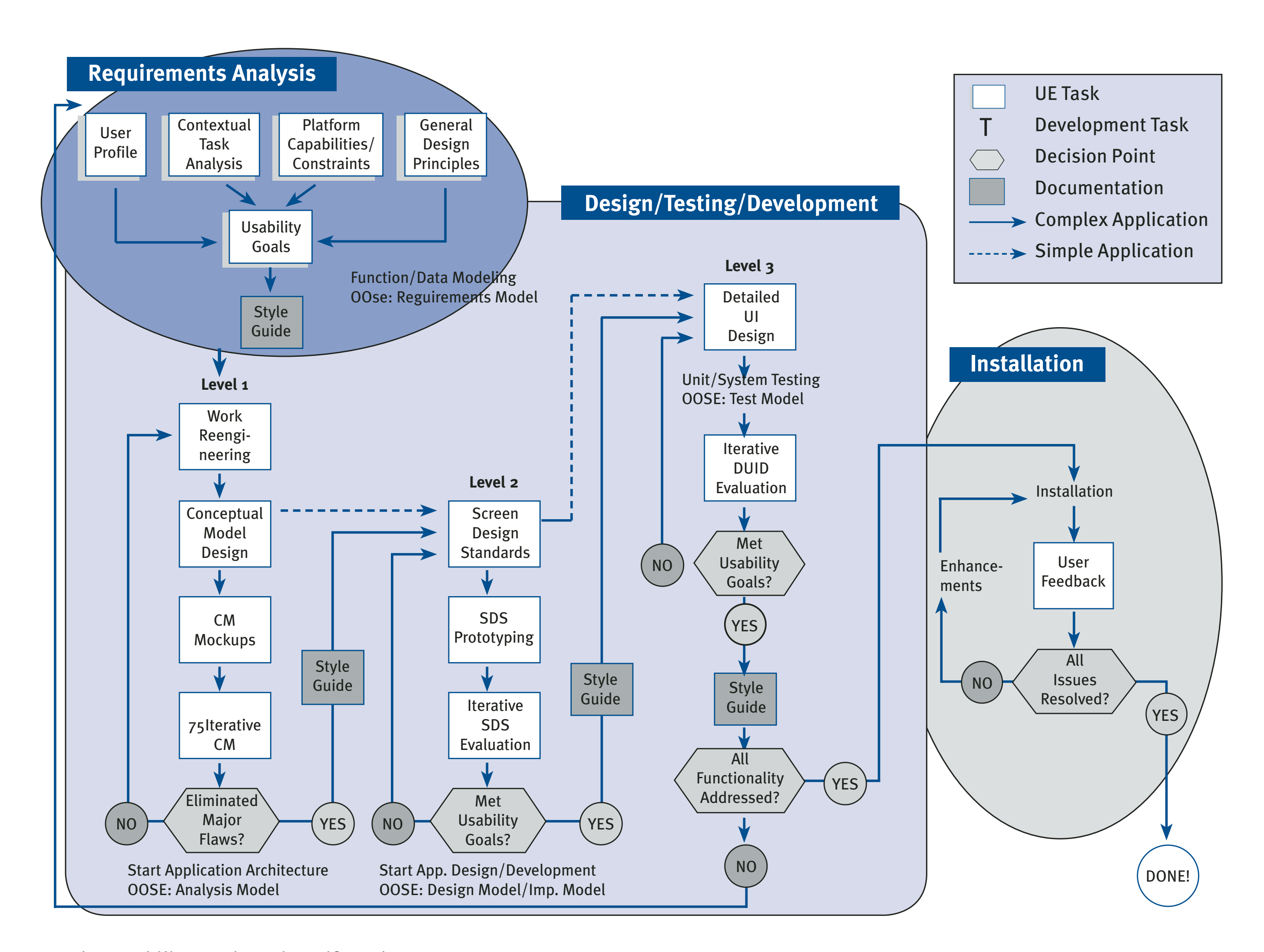
8.2 Benutzerzentrierter Entwicklungsprozess (DIN EN ISO 13407)

Die bisher vorgestellten Modelle sind Idealmodelle und werden in der Praxis so selten vollständig angewendet. Aber die Grundidee eines geplanten Vorgehens sollte in Software-Entwicklungsprojekten berücksichtigt werden. Man vereinbart Phasen (mit Zeitangaben), Aktivitäten, die in diesen Phasen stattfinden, und Ergebnisse (u. a. Code oder Dokumente), die zu einem bestimmten Zeitpunkt vorliegen müssen. Und in dieses Vorgehen (wie auch immer es benannt ist) sind nun die Aktivitäten des benutzerzentrierten Entwicklungsprozesses (Usability Engineering Lifecycle) zu integrieren. Es handelt sich also um kein konkurrierendes Modell.

Deutlich sichtbar ist hier die Unterscheidung zwischen einer Analysephase (Requirement Analysis) und einer Design-/Entwicklungsphase (Design/Testing/Development). Das besondere ist, dass der Usability Life Cycle eine dritte Phase, die so genannte Nutzungsphase, beinhaltet (Installation). Hier wird davon ausgegangen, dass sich trotz sorgfältiger Analyse und Entwicklung einige Nutzungsprobleme erst im „Echtbetrieb“ herausstellen.

Zwar ist es das Ziel, das Produkt fehlerfrei und mangelfrei auszuliefern. Die Realität zeigt aber, dass ein Software-Produkt, solange es im Einsatz ist, gepflegt/gewartet werden muss. Betrachtet man einmal näher was dort passiert, z. B. dass Funktionalitäten nachimplementiert werden, die so nicht vereinbart waren, sich aber als notwendig herausstellen. Unvorhergesehende Situationen führen z. B. zu unakzeptablem Antwortzeitverhalten, die Benutzer wollen Datenkombinationen eingeben, die nicht geplant waren, die Benutzer verursachen durch „Falscheingaben" Systemabstürze usw.

Diese Geschehnisse gilt es, von vorneherein in das Vorgehen mit einzuplanen. Dies erfordert, softwaretechnisch ein Versionen-Management und fachlich ein „Change-Management"-Verfahren zu installieren.
Deshalb ist für diese erste Nutzungsphase einer Software, deren Länge man z. B. auf 3 Monate festlegen kann, eine Sammlung und Bewertung des Feedback der Benutzer und entsprechende Nachprogrammierung vorzusehen. Damit hier keine Konkurrenzreaktion auftritt, welche Anforderungen die wichtigsten sind, sollte zu Beginn einer Software-Entwicklung ein Verfahren zur Klassifizierung der Fehler und Mängel vereinbart und entsprechend dokumentiert werden. Innerhalb der Entwicklungsphase finden sich drei Ebenen (Level). Die Ebene 2 und 3 kann man mit Grob- und Feinkonzept übersetzen. Die Ebene 1 beinhaltet die Analyse und Neukonzeption der Arbeitsabläufe. Ein optimaler Prozess beginnt mit einer solchen Phase.



The Usability Engineering Lifecycle

Quelle: D. Mayhew, The Usabillity Engineering Lifecycle, 1999

In der Praxis ist es durchaus denkbar, dass diese Phase ausdrücklich ausgeschlossen ist, d.h., dass nur die vorhandenen Aufgaben analysiert und dann 1:1 softwaretechnisch unterstützt werden sollen. Dann würde man mit dem Level 2 (Grobdesign) beginnen.

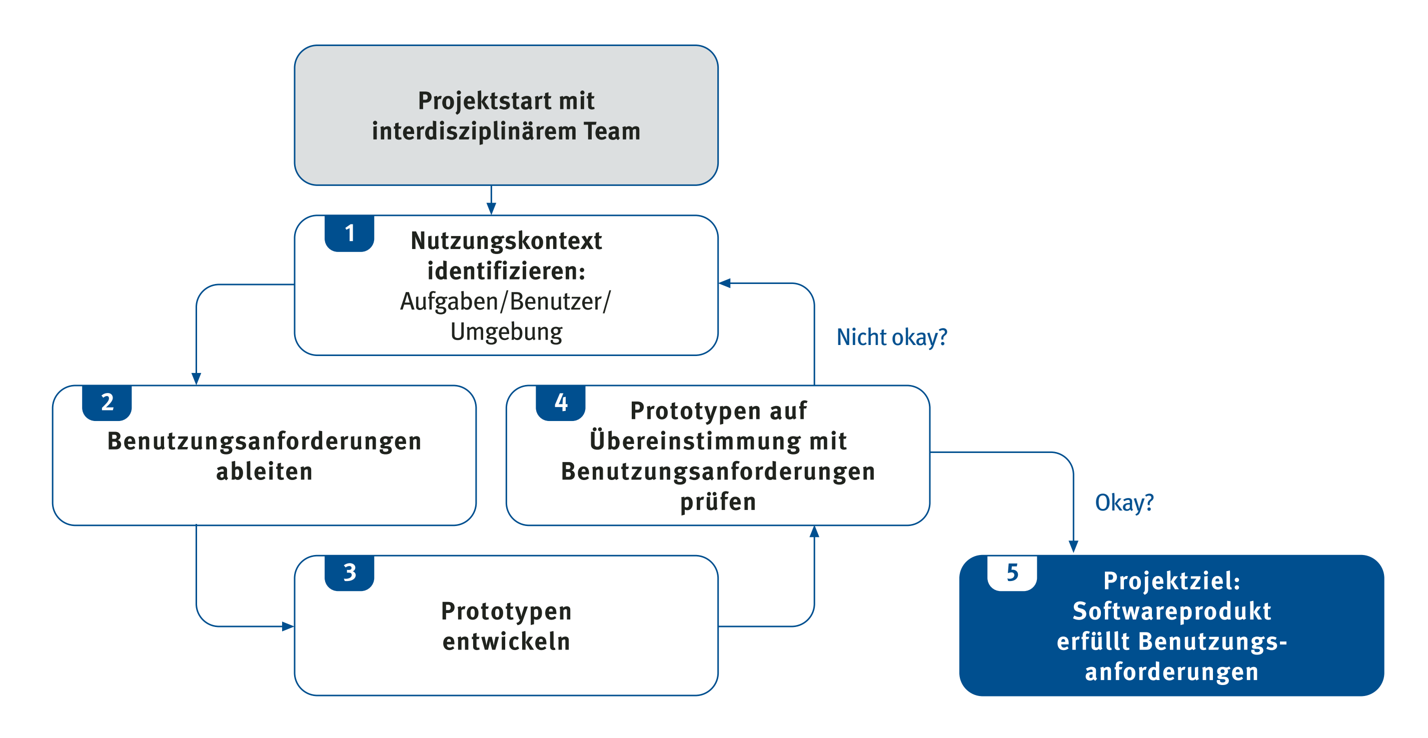
Auf dem Level 2 werden Standards für die Benutzungsschnittstellen entworfen und bewertet. Diese können sich auf Grundzüge des Maskendesigns, das Fensterkonzept, die Menü- und Navigationsstruktur beziehen und anhand von Prototypen realisiert werden.
Im Level 3 schließlich geht es um Details einzelner Masken. Standards werden mit Hilfe von Prototypen realisiert.

Während des gesamten Prozesses wird der Styleguide[3] fortgeschrieben. Das Modell sieht auch vor, je nachdem wie komplex ein Projekt ist, bestimmte Aktivitäten auszulassen. In vereinfachter Form findet sich dieses Vorgehensmodell in der DIN EN ISO 13407 wieder.

Fasst man diejenigen Aspekte zusammen, die einen wesentlichen Beitrag dazu leisten, dass ein Software-Produkt eine gute (ergonomische) Nutzungsqualität hat, sind folgende Aspekte herauszustellen (DIN EN ISO 13407):

1. Projektorganisation in interdisziplinären Teams
Die Projektorganisation erfordert Planungs- und Management-Kompetenz, die Zusammenarbeit mit verschiedenen Fachabteilungen und anderen Beteiligten sowie hohe kommunikative Fähigkeiten.

2. Analyse des Nutzungskontextes (Benutzer, Aufgaben, Umgebung)
Zum Nutzungskontext gehören neben der Arbeitsaufgabe, die Benutzer und die Umgebung, in der das Software-Produkt eingesetzt wird.
Prinzipiell ist es denkbar, dass eine Anwendung zwar für eine Aufgabe geeignet ist, aber nicht optimal in der Arbeitsumgebung funktioniert (z. B. ständiger Kundenkontakt erfordert hohe Flexibilität, telefonische Anfragen erfordern schnelle Antwortzeiten, akustische Rückmeldungen stören oder werden nicht gehört).
Ebenso ist denkbar, dass eine Anwendung, mit der überwiegend ältere Benutzer arbeiten, zu kleine Schriftzeichen verwendet, die nur sehr anstrengend gelesen werden können.
Es muss auch bekannt sein, welche und wie viele Benutzer mit der Software arbeiten werden, welche beruflichen Erfahrungen sie haben, welche IT-Kenntnisse vorhanden sind, wie oft sie die Anwendung nutzen werden, ob sie im Wechsel mit (welchen?) anderen Programmen benutzt wird, welche vor- und nachgelagerten Arbeiten mit welchen Personen abzustimmen sind u. ä.



Benutzerzentrierter Entwicklungsprozess

[3] Styleguide: Regelwerk zur Gestaltung der Oberfläche

Verstehen des Nutzungskontextes, das ist durchaus z. B. mit dem Berufsfeld der Ethnographen [4] vergleichbar. Ethnographische Leitlinien sind:

- Begegnungen gelten als wichtigste Grundlage zum Verstehen.
- Menschliches Verhalten ist nur unter Betrachtung des Umfeldes zu erklären.
- Man sollte sich die Sichtweise der Untersuchten aneignen.

Es bietet sich hierzu die Erhebung eines Kontextszenarios und mehrerer Use-Szenarien an.[5] Use-Szenarien unterstützen die Umsetzung von Tätigkeitsanforderungen (Benutzertätigkeit) in Produktmerkmale (Prototyp).

Ergänzend bietet sich eine Dokumentenanalyse an, wobei auch inoffizielle Dokumente (z. B. Merkzettel von Beschäftigten) Aufschluss darüber geben, wie die Beschäftigten sich ihre Arbeit organisieren.

Die aufgenommenen Informationen gilt es zu validieren, da es z. B. hierzu geschäftspolitische Vorgaben geben kann oder Aussagen verschiedener Fachabteilungen nicht übereinstimmen.

3. Anforderungen entwickeln und ableiten

Anforderungen entwickeln bedeutet zum einen, dass diese nicht von Anfang an eindeutig und vollständig feststehen oder durch eine Person definiert werden, sondern dass sich diese häufig erst durch eine Analyse der Arbeitsaufgabe herauskristallisieren. Dazu muss die eigentliche Arbeitsaufgabe gut verstanden sein, um dann zu entscheiden, welche Aufgabenunterstützung sinnvoll ist. Erst danach werden dann die passenden technischen Merkmale gesucht.
Die Anforderungsermittlung ist von der Anforderungsdefinition zu unterscheiden. Leider wird in vielen Projekten dieser erste notwendige Schritt übersprungen bzw. die Definitionsphase wird mit der Analysephase verwechselt.
Im ersten Schritt wird ermittelt, was gewünscht ist, und im zweiten Schritt wird definiert, wie dies erreicht werden soll. Denkbar ist durchaus, dass nicht alle Teile eines Geschäftsprozesses technisch unterstützt werden, sondern nur Teile. Eine kontrastive Vorgehensweise ist zu empfehlen.
Das „Wozu“ sollte z. B. gerade bei der Konzeption von Web-Auftritten die erste Frage sein. Nicht „wie will man sich als Firma darstellen?“, sondern „wozu will man den Web-Auftritt nutzen?“, ist der Aspekt, um den sich die einzuleitenden Maßnahmen gruppieren.
Geht man davon aus, dass Benutzungsfreundlichkeit nur zu einem kleinen Teil von der grafischen Oberflächengestaltung einer Software abhängt und der entscheidende Aspekt die richtig verstandenen Arbeitsaufgaben sind, zu deren Unterstützung die Software eingesetzt werden soll, kommt man an einer sorgfältigen Analyse der Geschäftsprozesse/ Arbeitsabläufe nicht vorbei.

[4] Völkerkundler

[5] DATech (2006) Prüfhandbuch Gebrauchstauglichkeit, http://www.datech.de

Die Analyse der Geschäftsprozesse und Arbeitsabläufe ist kein triviales Problem, was jede Unternehmensberatung bestätigen wird. Zumal nach der Analyse des Ist-Zustandes in der Regel über ein Soll-Konzept zu diskutieren ist, d. h. Reorganisations-Know-how ist gefordert.
Bei einem „Elektrifizieren" des Ist-Zustandes ist ein Scheitern des Projektes nur eine Frage der Zeit.
In dieser ersten Phase geht es darum, die Bedürfnisse des Kunden gut zu verstehen, auch wenn die Kunden selbst dies gelegentlich nicht genau formulieren können.
Die Frage hier lautet nicht, „was für ein Software-System ist gewünscht?", sondern „wozu soll das System dienen?".
Diese Frage muss der Geschäftsleitung ebenso gestellt werden wie den Beschäftigten an den Arbeitsplätzen. Beide Aussagen müssen in Übereinstimmung gebracht werden.

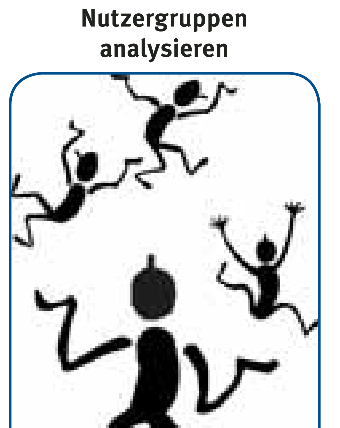
Anforderungen an das Dialogsystem sollen so formuliert werden, dass Arbeitstätigkeiten, angestrebte Arbeitsergebnisse oder Ziele zum Ausdruck kommen. In der Formulierung sind spezifische Merkmale zur technischen Realisierung (Implementierung) zu vermeiden.

Beispiele:
Schlecht (weil spezifische Lösungsmöglichkeit):
Das Dokument muss Verknüpfungen zu Anmerkungsdokumenten anbieten.
Gut (weil aus der Arbeitsaufgabe abgeleitet):
Der Benutzer muss im Dokument Anmerkungen machen können.

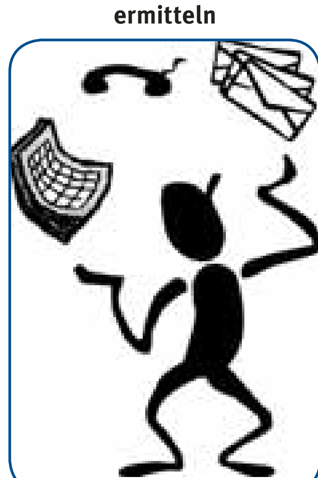
4. Entwicklung von Prototypen
Um die Umsetzung von Anforderungen zu demonstrieren, werden frühzeitig (ablauffähige) Modelle erstellt. Ein Prototyp ist die Darstellung einer Entwurfsidee, d. h., die Darstellung der ermittelten Anforderungen als Bildschirmoberfläche, Funktionalität o. ä. und bietet beiden Seiten (Hersteller und Anwender) Sicherheit. Prototypen können sich nur auf die Oberfläche beziehen oder eine Funktionalität durch alle Software-Ebenen realisieren.
Wichtig ist es, ein gemeinsames Verständnis gerade mit den Entwicklern darüber herzustellen, welche Art von Prototypen gemeint sind.
Die Prototypen werden in Workshops von Benutzern bewertet (anhand von Drehbüchern, die einen oder mehrere Arbeitsabläufe für den Prototypen festlegen). In moderierten Anwender-Entwickler-Workshops (½ Tag) sollten dann vorher festgelegte kritische Punkte besprochen werden.
Eine moderierte Anwender-Entwickler-Kommunikation bietet wichtige Vorteile:

- Unterschiedliche Begriffswelten werden angeglichen.
- Entwickler kommen auf neue Realisierungsideen, da ihnen der Anwendungsbereich vertrauter wird.
- Anwender machen neue Vorschläge, da sie die Möglichkeiten der Technik kennenlernen.

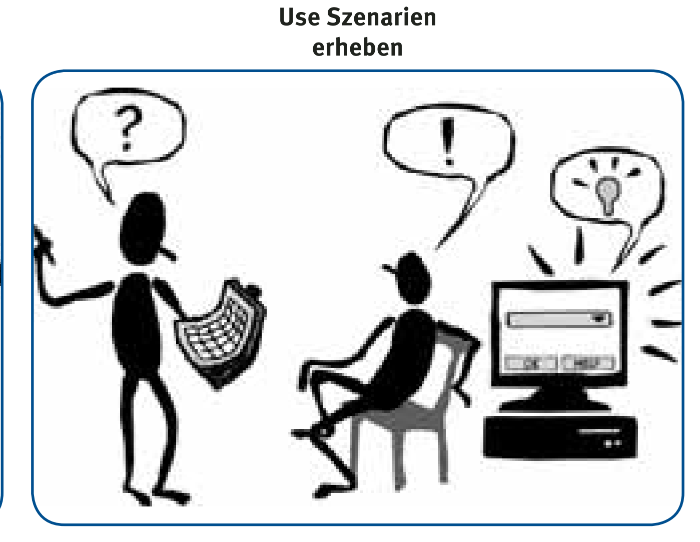
Nutzergruppen analysieren



Kontext ermitteln



Use Szenarien erheben



Zentrale Aspekte des User Centred Design

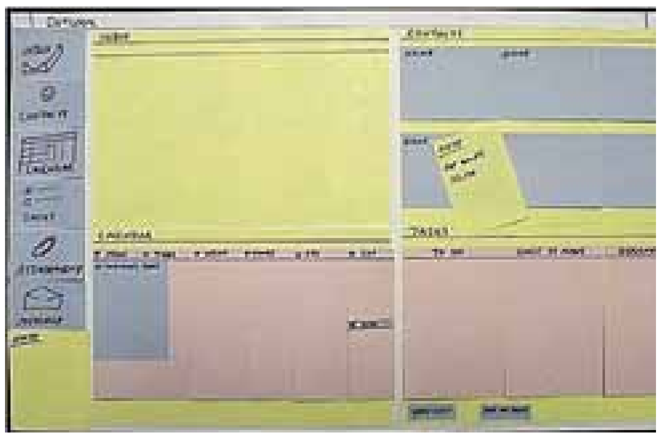
Dieses Vorgehen dient zur Erarbeitung eines gemeinsamen Verständnisses und zur Rückkoppelung übergreifender Aufgaben, es hilft, zusammen mit anderen Dokumenten, relevante Spezifikations- und Entwurfsprobleme zu klären.
Es dient zum Experimentieren und Sammeln von Erfahrungen und bietet eine Basis für Diskussionen und Entscheidungen. Vorab ist zu verabreden, welche Art von Prototyping im konkreten Projekt, in welchem Umfang gewollt ist.

Denkbar ist ein evolutionäres Vorgehen, in dem eine Ausgangssituation ständig weiterentwickelt wird, ein vergleichendes Prototyping, d. h. es werden mehrere Vergleichsprodukte bewertet, von denen die nicht gewollten anschließend verworfen werden. Rapid prototyping ist denkbar, wenn in schneller Folge neue Produkte zur Diskussion gestellt werden.
Dafür sind Bildschirmoberflächen ausreichend, die mit einem gängigen Tool (z. B. Power Point) schnell herzustellen sind. In diesem Stadium ist noch keine „Styleguide"-konforme Gestaltung der Oberflächen wichtig, sondern es soll herausgefunden werden, welche aufgabenrelevanten Daten auf welchen Masken in welcher Reihenfolge zur Verfügung stehen müssen. Dieses geht auch mit „Papier-Prototypen" (z. B. Post-It-Klebezettel auf einem Flipchart während eines Workshops).

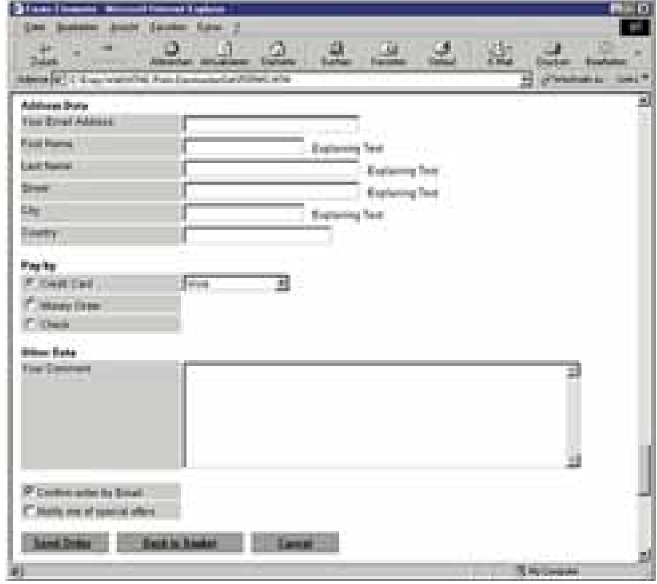


High Fidelity Prototyp
Prototyp, der den Großteil des Look und Feel des zukünftigen Produktes umfasst; beinhaltet oft essentielle Teile der Funktionalität und kann fast wie die reale Version benutzt werden. Der Grad, in dem das Layout und die Funktionen realisiert sind, hängt vom Zweck des Prototypen ab (Präsentation, Diskussion, Benutzertest usw.).

Eine gesteigerte Form des Prototyping ist, das Software-System in einer simulierten oder realen Einsatzumgebung zu testen. Wird z. B. ein Workflow-System eingeführt, das den gesamten Dokumentenfluss reorganiseren soll, ist es ratsam, nicht nur Software-Prototypen zur Veranschaulichung zu haben, sondern den Geschäftsvorfall quasi nachzuspielen. Vom Posteingang, über die Postverteilung, die Bearbeitung und den Postausgang. Es werden zahlreiche Fragen auftauchen, die nur über die Beurteilung der Oberflächen-Prototypen nicht hätten abgefangen werden können.
Komplexe Systeme werden häufig anders angewendet als geplant.



Low Fidelity Prototyp
Prototyp, der nur die konzeptuellen Aspekte umfasst. Dies kann in Papierform, als HTML-Version oder als Power-Point-Präsentation realisiert sein.



HTML-Prototyp, der nur aus Seiten besteht, die Screenshots mit Links enthalten, um die Navigationsaspekte einer Anwendung zu simulieren.

5. Evaluierung der Prototypen auf Übereinstimmung mit den spezifizierten ergonomischen Anforderungen unter Einbeziehung von (ggf. repräsentativen) Endbenutzern

6. Integriertes Testmanagament
Das klassische Testverständnis in Software-Entwicklungsprojekten beruht auf einer Verifikation, d. h. der Prüfung der Übereinstimmung zwischen der Anforderungsspezifikation und dem entwickelten Produkt. Die Anforderungsspezifikation ist aber die Beschreibung der technischen Umsetzung der fachlichen Anforderungen. Ob diese richtig verstanden wurden (Validierung), wird dabei dann nicht mehr geprüft.

Darüber hinaus ist zu bedenken, dass es quasi keine fehlerfreie Software gibt. Dies bedeutet, dass nur die Anwesenheit von Fehlern beweisbar ist, nie deren Abwesenheit. Ebenso sind fehlende Programmteile natürlich nicht testbar. Vielfach werden Werkzeuge (Software) zur Automatisierung eingesetzt. Automatische Tests sind jedoch nur für bestimmte Bereiche, Algorithmen und für kleinere Programmteile möglich. Die Korrektheit der Anforderungen und Designaspekte können nicht automatisiert getestet werden, sondern nur anhand von Dokumenten und/oder Prototypen.

In fachlichen Tests werden meist bestimmte repräsentative fachliche Funktionen mit definierten Testdaten herangezogen. Es wird geprüft, ob das Arbeitsergebnis effektiv ist, d. h. ob es korrekt und vollständig ausgeführt ist.
Wie diese fachlichen Funktionen im Arbeitsalltag dann von den Endbenutzern benutzt werden, ist oft schwer vorhersehbar. Da die Endanwender aber wertvolle Erfahrungen mit der alltäglichen Arbeit haben, sollte mit diesen Benutzern dann zusätzlich ein organisierter Usability Test stattfinden.
Beispiele für Alltagserfordernisse:

- Kunden erfragen bisher nicht angedachte Kombinationen von Vertragsklauseln.

- Ein Kunde hat nur eine Postfachadresse, aber die Eingabe der Straße ist Pflichtfeld.

- Der Sachbearbeiter soll telefonisch Auskunft über Tarife geben, vertippt sich und muss alle Angaben erneut vom Kunden abfragen.

- Der Sachbearbeiter ist mit dem Vorgang A beschäftigt und soll eine telefonische Auskunft zu Vorgang B geben.

Aus den o. g. Problemen ergibt sich, dass es erforderlich ist, ein Testkonzept zu erarbeiten, in dem genau festgelegt ist, was gegen was, wann, wer, womit und wie testet. Die festgelegten Schritte gilt es dann noch in das jeweils verwendete Vorgehensmodell zu integrieren. So kann z. B. in einem sehr frühen Stadium bereits ein Dokumententest stattfinden, in dem z. B. das Lastenheft mit dem Pflichtenheft abgeglichen wird. Die 5 W-Fragen des Testmanagements:

- Was? Objekte/ Aufgaben
- Gegen was? Fachliche, technische Anforderungen
- Wann? Stufen
- Wer? Entwickler, Benutzer ...
- Wie? Schleife ...
- Womit? Tools

Teststufe (Was?)	Gegen was?	Fokus auf.../Schicht	Beispiel	Wer
Funktionstest	DV-Konzept	Datenhaltung	Wird der Antrag mit der korrekten Prämie korrekt gespeichert?	Entwickler
GUI-Test	Oberflächenkonzept	Präsentationsebene/ GUI	Funktioniert die Auswahlliste?	Qualitätssicherung
Fachtest	Fachkonzept	Funktionen	Wird die Prämie korrekt berechnet?	Fachabteilung
Usability Test	Benutzer-Objekt-Modell	Use-Szenarien	Kann man erwartungskonform arbeiten?	Benutzer

Zur Frage, „was“ „gegen was“ getestet wird, sollten zumindest oben genannte Teststufen stattfinden: In der Praxis sind bei größeren Projekten weitere Teststufen erforderlich.

8.3 Methoden des Usability Engineerings

Never test the depth of the water with both feet.

Ausgewählte Mittel und Methoden (alphabetisch sortiert) zur Entwicklung von Nutzungskonzepten, Nutzungsanforderungen und Prototypen, deren Validierung, Bewertung und schrittweisen Verbesserung während des Entwurfs- und Pflegeprozesses.

Die Kernmethoden, die mindestens anzuwenden sind, wurden grau hinterlegt.

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Aufgaben-analyse	Methode der Arbeitswissenschaften, um Informationen über alle Einflussgrößen der Arbeit zu gewinnen, wie das organisatorische Umfeld, die Arbeitsaufgabe und -abläufe, Beziehungen zwischen Arbeitsaufgaben usw.	Untersuchung, ob Arbeitsaufgaben nach den Empfehlungen von DIN EN ISO 9241 Teil 2 gestaltet sind.	• Aufgabenanalyse und Aufgabendesign (oder Reengineering) sind zu unterscheiden. • Es ist sinnvoll, mit der Einführung einer neuen Software auch die bisherige Aufgabengestaltung zu analysieren und neu zu organisieren, um dann zusammen mit der Software z. B. eine Verkürzung von Durchlaufzeiten von Anträgen zu erreichen. Es gibt jedoch Projekte, bei denen die Neuausrichtung bisheriger Abläufe ausdrücklich ausgeschlossen ist.	Je nach eingesetzter Methode werden Informationen über die Einflussgrößen der Arbeit, wie das organisatorische Umfeld, die Arbeitsaufgaben und -abläufe, ermittelt.
Benutzer-befragung	Fragebogenaktion mit standardisierten Fragen, z. B. zur Dialoggestaltung.[6]	Erfassung der subjektiven Zufriedenheit und der Feststellung von Ursachen für Unzufriedenheit.	• Als Ergänzung zu anderen Methoden sinnvoll. • Es dürfen durch die Art der Fragestellung keine Verzerrungen entstehen (keine Suggestivfragen). Checklisten müssen auf ihre Relevanz überprüft werden (d. h. treffen alle Fragen zu?). • Der Auswertung der Benutzerbefragung sollte ein moderierter Benutzer-Entwickler-Workshop folgen mit dem Ziel kreativer Lösungsfindung für identifizierte Probleme.	Quantitative Daten (z. B. Prozent-Angaben) über Zufriedenheitsfaktoren
Benutzer-Entwickler Workshop	Eine kleine Zahl von Benutzern und Entwicklern trifft sich unter moderierter Anleitung, um für festgestellte Nutzungsprobleme Lösungen zu finden.	Zusammenbringen der Kenntnisse der Benutzer über das Anwendungsgebiet und effiziente Aufgabenbearbeitung und Kenntnisse der Entwickler über Entwicklungsmethoden und technische Lösungsmöglichkeiten.	Diese Methode kann nach einem Benutzertest oder einer Benutzerbefragung eingesetzt werden. Dient auch dazu, Kompromisslösungen zu finden, wenn die optimale Lösung nicht realisierbar ist.	Aufgabenangemessene Lösungsvorschläge

[6] z. B. „Ergonorm“-Benutzerfragebogen zu „Arbeit und Software“ (http://www.datech.de)

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Blickbewegungsuntersuchungen „Eye tracking“	Mittels Kameras werden die Augenbewegungen von Benutzern über eine oder mehrere Masken dokumentiert.	Visuelle Präferenzen (Aufmerksamkeitsverteilung) der Benutzer herausfinden.	Aufwendiges Verfahren, welches in Web-Anwendungen oft für Marketingaspekte angewandt wird (Herausfinden von Eye-catchern).	Quantitative Angaben, z. B.: Welche Elemente/ Seiten werden besonders häufig betrachtet.[7]
Dokumentenanalyse	Sichtung der vorhandenen Dokumente, z. B. Funktions- und Leistungsbeschreibungen, Anforderungsanalysen, Pflichtenhefte, Entwicklungsdokumentation, Beschreibungen des Fachkonzepts, Benutzerhandbuch, Formblätter u. ä.	Software-ergonomische Mängel aufdecken.	Insbesondere bei bereits existierenden Software-Produkten zu empfehlen. Analysekriterien sind auch Übersichtlichkeit und Verständlichkeit. Mangelnde Systematik in der Dokumentation ist oft ein Anzeichen für software-ergonomische Mängel.	Indizien für software-ergonomische Mängel.
Evaluation von ähnlichen Web Sites bzw. Wettbewerbern	Vergleich mit ähnlichen Web Sites anhand standardisierter Kategorien.	Prüfen und ggf. Übernehmen erfolgreicher Konzepte	Der Fokus des Vergleichs kann aus verschiedenen Perspektiven erfolgen, z. B. welche Inhalte werden angeboten, wie hoch ist die Nutzungsqualität, Design-Aspekte usw.	Auflistung einer Reihe von Kriterien wie Look und Feel, Navigation, Suchfunktion und deren Realisierungsform auf den verschiedenen Sites.
Experten-Inspektion	Ein Experte erstellt eine Mängelliste anhand vorher abgesprochener Kategorien.	Aufdecken software-ergonomischer Mängel (z. B. Schriftgröße) ohne zwingenden Bezug zur Aufgabe.	Ist als Ergänzung zu Benutzertests sinnvoll, allein jedoch nicht ausreichend. Ein Experte findet jedoch auch zusätzliche Mängel heraus, die Benutzer nicht zwangsläufig identifizieren.	Detaillierte Mängelliste, ggf. bereits mit Prioritäten und Redesign-Vorschlägen.
Focus Group/ Workshop	Eine Fokus-Gruppe bringt verschiedene Beteiligte zu einer Art informeller Diskussionsrunde zusammen. Die verschiedenen Sichten, die für das Software-Produkt relevant sind, werden „herausgelockt“.	Identifizieren erster Anforderungen, z. B. an eine Intranetseite. Kann auch als Feedback für ein bereits eingesetztes System genutzt werden.	Von Web-Agenturen verwendete Methode, deren Ziel in erster Linie Ideenfindung ist. Deshalb sollte diese Methode nur ergänzend eingesetzt werden. Es ist auf eine repräsentative Auswahl der Teilnehmer zu achten.	Qualitatives Feedback.
Heuristische Evaluationen	Eine kleine Zahl von Gutachtern untersucht die erstellte Software und überprüft, inwieweit diese mit ausgewählten Usability-Prinzipien (Heuristiken) übereinstimmt.	Kontextunabhängige Bewertung.	Im Gegensatz zu den aus dem Nutzungskontext, Benutzerbelangen und Aufgaben abgeleiteten Anforderungen sind Heuristiken kontextneutral. Dies bedeutet, dass eine Heuristik, die für ein Softwareprodukt passt, bei einem anderen nicht ausreicht.	Zusammenstellung eines ersten Eindrucks der Nutzungsqualität.

[7] Anmerkung zu automatisierten Methoden, wie sie z. B. von Web-Agenturen angeboten werden: Clickstream-Analyse: Bei dieser Methode wird die Anzahl der Mausklicks gemessen, die Benutzer benötigen, um z. B. auf einer Web-Site zu einem Produkt zu navigieren. Diese Zahl ist nicht aussagekräftig zur Feststellung der Usability. Ein Programm kann nur die Zeit für einen optimalen Weg berechnen. Ein Programm kann nicht feststellen, ob ein Klick sinnvoll war oder nicht. Die reale Usability hängt von dem Weg ab, den der Nutzer geht, und er geht nicht immer den optimalen Weg. Deswegen ist wichtiger, wie er sich zurechtfindet, wenn er sich verlaufen hat. Eine Site kann sogar gebrauchstauglicher sein, obwohl der User einen Klick mehr benötigt. Z. B. indem der Benutzer schneller weiß, wo er ist, sich besser und gezielter orientieren kann. Automatisiert kann die Antwortzeit, die Aktualität der Links u.ä. getestet werden.

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Karten-sortierung	Die Substantive und Verben, die aus den Use-Szenarien identifiziert werden können (z. B. „Angebot abgeben“), werden auf kleine Karten geschrieben und werden von Benutzern in eine für sie logische Struktur gelegt.	Menüstrukturen entwerfen. Zuordnung von Objekten und Aktionen aus Benutzersicht.	Mit einer repräsentativen Benutzergruppe werden hohe qualitative Ergebnisse erzielt.	Designvorschläge.
Nutzungs-kontext (Umfeld-) Analyse	Untersuchung des Nutzungskontextes, d. h. der Faktoren, die die Gebrauchstauglichkeit eines Produktes beeinflussen. Dies sind neben den Benutzern und den Aufgaben auch physikalische, soziale und organisatorische Umgebungsfaktoren (z. B. Home vs. Office, geöffnetes vs. geschlossenes Büro, laut vs. leise, hoher vs. niedriger Unterbrechungsgrad, Helligkeit) bis hin zum Niveau der Schulungsunterstützung.	Verständnis der Erfordernisse der Arbeit und der Organisation in ihren komplexen Zusammenhängen für den erwarteten Benutzungskontext. Das Ergebnis bestimmt die Usability-Ziele und wirkt sich auf die Gestaltung der Benutzungsschnittstelle aus.	Neben Abläufen zwischen Abteilungen, der organisatorischen Umgebung und der physikalischen Umgebung spielen auch technische Rahmenbedingungen eine Rolle für die Usability: Größe des Bildschirms, Bildschirmauflösung, darstellbare Farbtiefe, Rechnerleistung, Größe des verfügbaren Arbeitsspeichers, verwandtes Netzwerk etc. In Web-Applikationen spielen u. a. die verwendeten Browser der Nutzergruppen eine Rolle usw.	Arbeitsdokument, das im Laufe der Entwicklung vervollständigt wird.
Kontext-Szenario	Episodische Beschreibung von Aufgaben und Tätigkeiten in ihrem Kontext. Die Beschreibung erfolgt ohne konkreten Bezug zu Software-Produktmerkmalen. Die Beschreibung hat nicht den Anspruch auf Vollständigkeit.	Dient der Ermittlung der Aufgabenerfordernisse und der Ableitung von Dialoganforderungen.	Zur Durchführung empfiehlt es sich, Leitfragen vorzubereiten. Die sachliche Richtigkeit und Vollständigkeit des Szenarios muss von den Benutzern bestätigt werden (Validierung). Die Erhebung sollte nicht am Arbeitsplatz stattfinden, sondern an einem Ort, an dem man mindestens 30 Minuten ungestört ist.	Inhalt: • allgemeine Arbeitsabläufe, • Voraussetzungen für die Bearbeitung, • normale Reihenfolge der Durchführung, • Besonderheiten. Darüber hinaus wird der Nutzungskontext beschrieben (physikalische Arbeitsumgebung und technologische Parameter), um die technischen Beschränkungen und Möglichkeiten für die Gestaltung der Benutzungsschnittstelle zu erfassen.
Konzept-Workshop	Von Web-Agenturen häufig eingesetzte Methode, um mit den Beteiligten Navigationsmodell und visuelles Konzept abzustimmen.	Verbindliches Grobkonzept für die Entwurfsphase.	Das Navigationsmodell umfasst die Site-Map, den Navigationsrahmen und Storyboards für die Navigation sowie die Fensterspezifikationen.	Grobkonzept für die Präsentation und Navigation einer www-Applikation.

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Kreativ-Workshop	Workshop zum Ermitteln der Designpräferenzen, des Markt-Images, der Corporate Identity und ähnlicher Faktoren.	Definieren des visuellen Designs.	Bei www-Applikationen kann es sinnvoll sein, Kreativ- und Usability-Workshops getrennt durchzuführen, um beide Aspekte angemessen zu identifizieren. Kreativ-Workshops sensibilisieren lediglich für das Thema Usability. Für jeden Bereich sollten Verantwortlichkeiten festgelegt werden.	Textuelle Beschreibung des gewünschten visuellen Erscheinungsbilds der Anwendung. Hinzu kommen Beispiele von „Branding“-Materialien, vergleichbaren visuellen Designs u.ä.
Nutzergruppenanalyse	Beschreibung der potenziellen Benutzergruppen der Software mit Bezeichnung ihrer Position, Ausbildung, Nutzungshäufigkeit, ihrer IT-Erfahrung usw. Ergänzt wird dies durch Angaben zu Anzahl, Alter, Geschlecht, Nationalität u.ä.	Für die definierten Benutzergruppen werden Use-Szenarien ermittelt und die Dialogabläufe für diese Szenarien optimiert.	Diese relativ trivial erscheinende Methode wird in der Praxis oft vernachlässigt. Da eine Software aber nicht für jeden potenziellen Benutzer und beliebige Aufgaben gleichzeitig optimiert werden kann, ist die Priorisierung von Benutzergruppen Voraussetzung für eine effiziente Entwicklung.	Textuelle Beschreibung der Nutzerguppen als Teil des Usability Konzeptes.
Prototyping	Ein iterativer, d. h. sich wiederholender Prozess zur Herstellung einer Benutzungsoberfläche anhand von Use-Szenarien, sowie zur Herstellung oder Simulation einer Funktionalität.	Veranschaulichung der ermittelten Benutzeranforderungen, so wie sie der Entwickler verstanden hat.	Neben der Nutzung von entsprechender Software kann man auch mit Papier und Bleistift auf Wandzeitungspapier arbeiten. Dies hat den Vorteil, dass man sich nicht in technische oder Oberflächendetails verliert, sondern sich auf die wesentliche Ablauffolge konzentriert. Auch mit Power Point können relativ einfach Prototypen erstellt werden.	Der Prototyp ist die Darstellung einer Entwurfsidee, das heißt die Umsetzung von Anforderungen der Benutzer in Produktmerkmale. Simulation eines gewünschten Dialogablaufs und/oder einer dahinterliegenden Funktionalität.
Teilnehmende Beobachtung	Die Erfassung des Tätigkeitsverlaufs bei einem Benutzer mit den Wechselwirkungen zwischen Verlauf und Nutzungskontext (durch stilles Danebensitzen und Erfassen der Vorkommnisse).	Dient der Erfassung von Merkmalen einer Tätigkeit im Zeitverlauf (z. B. zögerlicher Ablauf, Nachschlagen, Rückfragen, umständlicher Ablauf...). Es geht hier nicht um Software-Produktmerkmale.	Der Benutzer soll durch diese Beobachtung nicht beeinflusst werden. Der Einsatz von Video-Aufzeichnungen ist hier theoretisch möglich, aber in der Praxis selten zu empfehlen. Ergänzt werden kann diese Methode von einem halbstrukturierten Interview, d. h. der Interviewer bereitet zwar eine grobe Fragestruktur vor, lässt aber überwiegend den Benutzer erzählen.	Dokumentation insbesondere der kritischen Merkmale oder Tätigkeiten im Arbeitsablauf.

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Usability-Workshop	Treffen mit den entsprechenden Beteiligten am Beginn eines Projektes, bei dem Nutzergruppen, der Nutzungskontext und die zu unterstützenden Use-Szenarien und daraus resultierende Usability-Ziele festgelegt werden.	Definieren von nachweisbaren Usability-Zielen für die Integration in ein Qualitätskonzept.	In Web-Projekten (z. B. Intranet) kann diese Methode zusammen mit dem Kreativ-Workshop die Trennung und Zusammenarbeit der beiden Bereiche festlegen.	Das Usability-Konzept beinhaltet eine Beschreibung der Nutzergruppen, des Nutzungskontextes und die zu unterstützenden Use-Szenarien. Ergänzend werden gewünschte, daraus resultierende Usability-Ziele beschrieben.
Usability-Test mit Benutzern	Eine Gruppe repräsentativer Nutzer bearbeitet ausgewählte Testaufgaben in einem definierten Zeitraum.	Feststellen der „harten“ softwareergonomischen Probleme.	Erfahrungen zeigen, dass bis maximal sechs Testpersonen ausreichen, um 80 % der Schwachstellen des Systems zu entdecken. Bei der Auswahl von Testpersonen sollte berücksichtigt werden: Geübte und ungeübte Benutzer, gemischte Altersstruktur, weibliche und männliche Personen, Personen, die der Software-Einführung kritisch gegenüber stehen, als auch Befürworter. Voraussetzung ist, dass das System stabil läuft, die Testdaten funktionieren, so dass sich die Benutzer auf die Ausführung der Aufgabe konzentrieren können.	Mängelliste, die anschließend priorisiert wird. Für die Abarbeitung der Mängel müssen Kapazitäten eingeplant werden.
Use-Szenarien	Episodische Beschreibung von Arbeitstätigkeiten unter Berücksichtigung des Nutzungskontextes bzw. der tatsächlichen Ausführung der Aufgabe.	Use-Szenarien dienen der Erfassung der Interaktion des Benutzers mit der Software unter den gegebenen Nutzungsbedingungen am Bildschirmarbeitsplatz. Ziel ist das Identifizieren von Hauptaufgaben und sich daraus ableitenden Anforderungen an die Software. Web-Anwendungen können jeweils nur für bestimmte Use Szenarien optimiert werden.	Es wird der normale Arbeitsablauf beschrieben. Der Titel sollte aus einem Substantiv in Verbindung mit einem Verb bestehen (z.B. Antrag anlegen). In der Praxis gilt es, möglichst viele Use-Szenarien zu sammeln, um diese dann zu sortieren, zu gruppieren und zu priorisieren. In jedem Fall werden die Szenarien mit Benutzern validiert, d. h. sie bekommen den schriftlichen Entwurf und müssen deren Korrektheit bestätigen. Daraus können bereits erste Benutzer-Objekte und Aktionsbezeichnungen für die Menüstruktur abgeleitet werden. Die objektorientierte Entwicklung kennt sog. Use Cases (Anwendungsfälle). Ein Use Case ist dort die Bschreibung einer Menge von Systemaktivitäten aus der Sicht der Akteure, die zu einem wahrnehmbaren Ergebnis führen. Dies sind tabellarisch verkürzte Formen von Use Szenarien und dienen dazu, die dynamische Sicht der Arbeit zu beschreiben (in Ergänzung zum statischen Klassen-Modell). Sie ersetzen jedoch nicht die Use Szenarien.	Textuelle prosaische Beschreibung der Haupttätigkeitsabläufe (max. 1 Seite pro Vorgang), die Teil des Usability-Konzeptes ist. Ein protokolliertes Use-Szenario ist während der Erhebung ein noch relativ ungeordnetes Dokument. Für den beschriebenen Tätigkeitsablauf muss eine (ggf. vom Ablauf im Alltag idealisierende) Ablaufstruktur der Aufgabenschritte gefunden werden. Für den Fall, dass es um die ergonomische Bewertung einer Aufgabe geht, ist eine Illustration kritischer Nutzungssituationen mittels „screen shots” nützlich, um die beschriebenen Nutzungsprobleme besser zu veranschaulichen.

Methode	Beschreibung	Ziel	Ergebnis	Hinweise
Styleguides	Beschreibt das Standardverhalten der grafischen Objekte einer Software und enthält Empfehlungen zu deren Einsatz. Oft interner Standard bei Software-Herstellern oder Unternehmen.	Konsistente Gestaltung der Oberflächenmerkmale auch über verschiedene Anwendungsprogramme hinweg oder innerhalb einer Anwendung.	Oft beinhalten Styleguides nur Festlegungen zur Verwendung von Oberflächenelementen. Es ist zu empfehlen, dies um Festlegungen zum Fensterkonzept, zur Dialoggestaltung, zu Interaktionstechniken etc. zu ergänzen. Lebendiges Werk, das fortlaufend erweitert bzw. angepasst wird Ausnahmen vom Styleguide sind zulässig, müssen aber begründet werden.	Rahmen für die Qualitätssicherung, Nachschlagewerk für die Software-Weiterentwicklung, aber auch für neue Teammitglieder.
Trainings	Software-Ergonomie-Trainings für Entwickler.	Kenntnisse über grundlegende Gestaltungsregeln bzw. Anwendung ergonomischer Normen.	Es ist sinnvoll, dies projektspezifisch zu Beginn anhand der zu entwickelnden Software zu veranstalten. Nach einer Einführung kann so bereits am konkreten Entwurf gearbeitet werden.	Geschulte Entwickler, die kompetent mit den Benutzern Entscheidungen treffen können.

8.4 Beurteilung/Messung der Gebrauchstauglichkeit (DIN EN ISO 9241-11)[8]

In der DIN EN ISO 9241-11 finden sich Erläuterungen, wie Gebrauchstauglichkeit zu bewerten ist. Als Messkriterien für gebrauchstaugliche Software sind dort definiert:

Kriterium	Definition	Beispiel für Messgröße
Effektivität	Grad der Erreichung angestrebter Ziele (Zielerreichung und Qualität des Arbeitsergebnisses)	Erfassung der Richtigkeit/Vollständigkeit des erzielten Arbeitsergebnisses. Grad der Zielerreichung in Prozent = Antrag zu 80 % ausgefüllt
Effizienz	Aufwand zur Erreichung der definierten Ziele	Erfassung der Bearbeitungszeit einer Aufgabe oder für eine bestimmte Anzahl von Dialogschritten, Korrekturzeiten
Zufriedenstellung	Grad der Zufriedenheit	Befragung der Benutzer Subjektive Bewertung auf Skalen

Messkriterien zur Feststellung der Gebrauchstauglichkeit von Software

Grundsätzlich kann man Messungen quantitativ oder qualitativ angehen.
Quantitative Ansätze gehen davon aus, dass Benutzbarkeit nur dann gegeben ist, wenn quantitative Zielvorgaben erfüllt sind, wie vergleichbare andere Ziele (z. B. Kostengünstigkeit).
Der quantitative Ansatz erfordert eine Festlegung von Attributen für „Benutzbarkeit“, z. B. erfolgreiche anfängliche Benutzung, Lernfortschritte, Schwierigkeiten bei unregelmäßiger Benutzung. Auch ist eine Entscheidung notwendig, woran ein Attribut gemessen werden soll, z. B. Aufgabenbearbeitungszeit, Anzahl der Fehler.
Es können Zielwerte festgelegt werden (schlechtester und optimaler Fall). Es wird also bei der Entwicklung versucht, die Benutzbarkeit im definierten Sinne zu erreichen. Der Prozess wird als beendet betrachtet, wenn tolerable Werte (Attribute) erreicht sind.
Vorteil: Erreichen der festgelegten Werte ist ein Ergebnis, dass für alle sichtbar und messbar ist.

[8] streng genommen nur gültig für Teile 14–17

Der **qualitative Ansatz** geht davon aus, dass Benutzbarkeit nur durch eine schrittweise und rückgekoppelte Vorgehensweise erreicht wird.
Im Gegensatz zum quantitativen Ansatz geht man hier davon aus, dass gute Systeme nicht im ersten Anlauf entstehen. Die Vorgehensweise ist schrittweise und partizipativ. Probleme der Benutzer werden nicht nach vorgegebenen Kategorien erfasst, sondern individuell beobachtet. Dies erfordert geübte Beobachter. Testpersonen werden aufgefordert, bei der Aufgabenbearbeitung laut zu denken. Es werden semi-strukturierte Interviews geführt, es werden Bildschirmaufzeichnungen gemacht und diese den Entwicklern vorgeführt. Das System wird durch andere Entwickler kritisch inspiziert.

Bei größeren Projekten kann man ergänzend quantitative Werte festlegen.

Welche Methoden man auch anwendet, relevant ist, dass sie folgenden Ansprüchen der Praxis genügen:

- Transparenz, d. h. die Methode und Messverfahren müssen nachvollziehbar sein.
- Praktikabilität, d. h. der Aufwand muss in einem angemessenen Verhältnis zum Nutzen stehen.
- Flexibilität, d. h. bei Bedarf sollte man Personen, Zeit, Durchführung variieren können.
- Konsensfähig, d. h. Entwickler und Benutzer sollten sie akzeptieren.
- Ergebnisorientiert, d. h. die Ergebnisse sollten konstruktive Vorschläge für die Weiterentwicklung liefern.

DATech Prüfverfahren [9]
In jedem Bereich, in dem es Normen gibt, gibt es auch vorgeschriebene Prüfverfahren zur Einhaltung dieser Normen, um die Objektivität zu gewährleisten. In der DIN EN ISO 9241 Teil 11 ist beschrieben, wie man die Anforderungen aus den Teilen 14–17 der Norm spezifizieren und messen kann.
Die DIN EN ISO 9241 Teil 10 (Dialoggestaltung) enthält keine Prüfvorschrift, so dass eine Prüfung grundsätzlich Vereinbarungssache ist.

Kurz skizziert ist die Grundidee, dass aus der DIN EN ISO Teil 10 Prüfkriterien (Mindestanforderungen) abgeleitet werden und geprüft wird, ob diese **nicht** eingehalten werden. Ist dies der Fall, wird zunächst eine Normabweichung vermutet. Dann wird beurteilt, welche Auswirkung diese Normabweichung hat. Im nächsten Schritt wird beurteilt, ob die Auswirkung bedeutsam ist und ob der Mangel behoben, gelindert oder kompensiert werden kann. Erst wenn dies **nicht** möglich ist, wird ein Normenverstoß festgestellt. Gleichzeitig sollte jedoch ein konstruktiver alternativer Gestaltungsvorschlag gemacht werden, für den man sich die Anregungen aus den Normteilen DIN EN ISO 9241, Teile 12–17 sucht.

Für die DIN EN ISO 9241-10 gibt es einen Leitfaden, wie ein Softwareprodukt auf Normkonformität mit DIN EN ISO 9241, Teile 10 und 11, zu prüfen ist. Dieser Leitfaden[10] dient Prüfern dazu, Prüfkriterien zu entwickeln.

Zur Durchführung einer solchen Prüfung sind nur akkreditierte Stellen, wie z. B. der TÜV oder andere Usability Labore, befugt. Als letztes Mittel in einem Streitfall kann man sich hier also ggf. bestätigen lassen, dass eine Software gebrauchstauglich ist oder eben auch nicht.

Sinnvoller ist es in jedem Fall, den konstruktiven Dialog im Unternehmen zu suchen und z. B. über die Qualitätssicherung ergonomische Kriterien als verbindliche Richtlinien zu vereinbaren.

---

[9] DATech Deutsche Akkreditierungsstelle Technik e. V. (benennt Gutachter, akkreditiert und überwacht Prüfinstitute wie z. B. den TÜV)

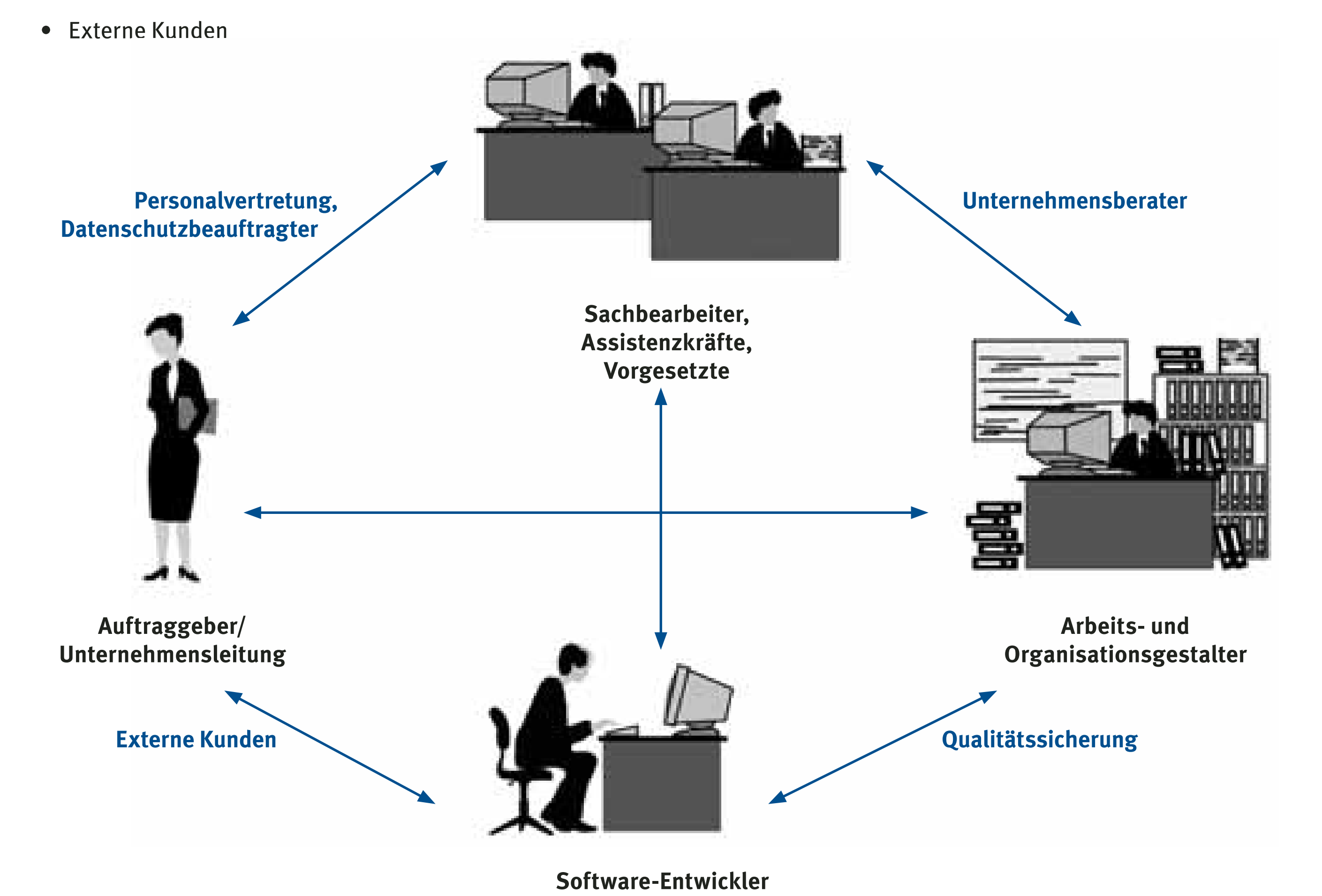
[10] DATech (2006) Prüfhandbuch Gebrauchstauglichkeit http://www.datech.de

8.5 Kommunikation in Software-Entwicklungsprojekten

Die Erfahrungen zeigen, dass erfolgreicher Softwareeinsatz nicht in einem Schritt durch den Beschluss einiger Entscheider erfolgt, stattdessen handelt es sich stets um einen zyklischen Prozess, an dem darüber hinaus zahlreiche Personen mit unterschiedlichen und berechtigten Interessen, ggf. zu unterschiedlichen Zeitpunkten beteiligt sind. Diese Personen müssen oft unter Zeit- und Kostendruck Entscheidungen treffen.

Beteiligte in Software-Entwicklungsprojekten können sein:

- Auftraggeber / Unternehmensleitung
- Benutzer / Sachbearbeiter
- Entwickler
- Externe Kunden
- Unternehmensberatung
- Datenschutzbeauftragte
- Assistenzkräfte
- Vorgesetzte / Fachabteilungsleiter
- Arbeits- und Organisationsgestalter
- Qualitätssicherer / Controller
- Personalvertretung
- Externe



Kommunikationswege und -barrieren zwischen Beteiligten in Software-Entwicklungsprojekten

8 Der Software-Entwicklungsprozess

Software-ergonomische Verbesserungen lassen sich fast nie durch eine Konfrontation der Interessen durchsetzen, sondern erfahrungsgemäß am effektivsten, indem ein Konsens zwischen den Beteiligten gefunden wird. Oft sind es sogar Missverständnisse über Begriffe, die in der Folge zu einer unangemessenen Softwarefunktionalität führen.
Die organisierte Kommunikation zwischen diesen Beteiligten ist Voraussetzung für erfolgreiche Software-Entwicklung. Es gilt, Interessen zu erkennen, abzuwägen, Kompromisse zu finden. Für Abstimmungen oder Entscheidungen die zu treffen sind, sind Moderationsmethoden unerlässlich.

Eine darin geübte Person sorgt dafür, dass

- einzelne wichtige Interessen nicht übergangen werden,
- nicht nur die Aussagen der „Lautesten“ berücksichtigt werden,
- „Skeptiker“, die an Schlüsselpositionen sitzen, nicht übergangen werden,
- zuständige Personen (ggf. gegen ihre Absicht) einbezogen werden,
- nicht offen artikulierte Interessen erkannt werden,
- widersprüchliche Aussagen geklärt werden,
- zwischen kurz- und langfristigen Interessen unterschieden wird,
- persönliche, private und berufliche Interessen Sachentscheidungen nicht blockieren,
- Sachverhalte korrekt verstanden sind, da es nicht ungewöhnlich ist, dass Sachverhalte anders formuliert werden, als sie gemeint sind.

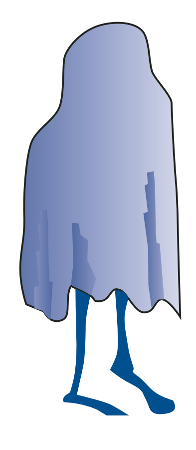
Die Liste ließe sich beliebig fortsetzen.

Moderierte Kommunikation, weil...

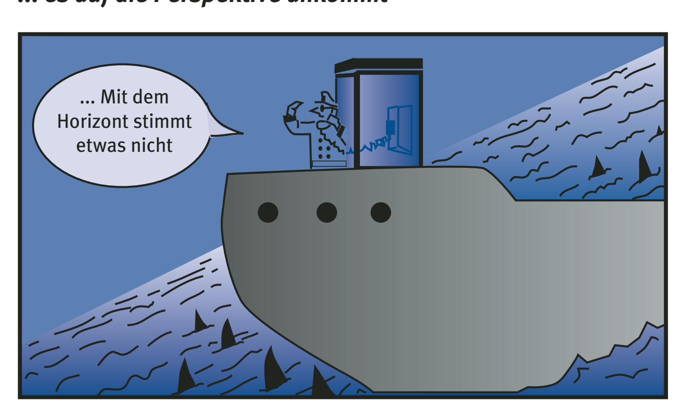
...Begriffe unterschiedlich definiert werden



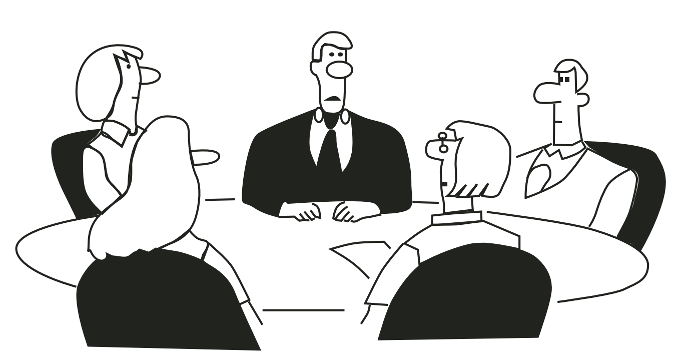
... nicht alles offen gesagt wird



... es auf die Perspektive ankommt



... dann Entscheidungen getroffen werden



Diese Aspekte sollten bei der Moderation der Anforderungsentwicklung beachtet werden, denn moderiert werden Menschen, nicht Themen. Gerade das Zusammenführen der verschiedenen Sichten auf einen Arbeitsablauf bzw. Geschäftsprozess macht den Erfolg eines entsprechenden Softwareproduktes aus.

Gerade im Bereich der so genannten „Informations- und Kommunikationstechniken“ liegt eine Reduzierung des Kommunikationsbegriffs auf die technischen Aspekte nahe. Dennoch wird die Technik von Menschen und für Menschen gestaltet. In diesem Sinne ist Kommunikation der verhaltensbeeinflussende (Informations)-austausch zwischen Menschen.
In der Kommunikationsforschung unterscheidet man zwischen der Sach- und der Beziehungsebene. Über 60 % menschlichen Handelns findet auf der Beziehungsebene statt. Dies gilt auch in Unternehmen. Dies gilt auch in Software-Entwicklungsprojekten. Oft ist es so, dass Konflikte auf der Beziehungsebene auf der Sachebene als Widerstand zurückkommen.

Fragt man verschiedene Menschen nach der Definition bzw. Unterschieden von folgenden Begriffen wird man höchst unterschiedliche Antworten erhalten.

Unterbrechbarkeit — Abbrechbarkeit,
Versicherungssumme — Deckungssumme,
Angebot — Antrag
Komponente — Modul
Courtage — Provision
Prototyping — Prototypen

Bei Watzlawick[11] findet sich ein gutes Beispiel:

Ein hochgelehrtes Kollegium von englischen Medizinern diskutierte lange über die Frage, ob in den Nerven ein „liquor“ fließe. Die Mediziner fanden zu keiner Einigung, bis jemand fragt, ob denn alle wüssten, was man unter „liquor“ verstünde. Die im ersten Moment provokativ anmutende Frage führte schließlich zu einem schnellen Schluss der Debatte. Es zeigte sich, dass die Debatte in der Wortbedeutung begründet war. Die eine Partei hatte unter „liquor“ eine reale Flüssigkeit (wie Wasser oder Blut) verstanden und bestritt demgemäss, dass so etwas in den Nerven fließe. Die andere Partei interpretierte das Wort in Richtung fluidum (wirkende Kraft, wie Elektrizität) und war deshalb überzeugt, dass ein „liquor“ durch die Nerven fließe.

Um eine Aufbau- und Ablauforganisation zu verstehen und um diese Strukturen in Software abzubilden, bedarf es hoher kommunikativer und sozialer Fähigkeiten. Zwar gibt es auf Papier (z. B. Organigramme) oder in den Köpfen einiger zuständiger Personen feste Vorstellungen davon, wie die „Arbeit laufen soll“. Neben den offiziellen gibt es aber immer auch inoffizielle Strukturen und Kommunikationswege.
Sowohl die Existenz dieser Strukturen als auch das Umgehen von Regeln scheint bis zu einem gewissen Grad zu bewirken, dass hierarchische Organisationen überhaupt funktionieren.

Software-Entwicklung bedeutet das gemeinsame Herausbilden eines gesicherten Verständnisses über die gewünschte Funktionalität und die Nutzungsmöglichkeiten der Software. Dies muss durch intensive Kommunikation entstehen und nicht über Dokumente und Datenbanken.[12]

In der DIN EN ISO 9241 Teil 2 steht dazu:

„Die erfolgreiche Pflege eines bildschirmgestützten Informationsverarbeitungssystems erfordert eine kontinuierliche Überprüfung der entsprechenden Kriterien.
Effektive Kommunikationsformen sollten unterhalten werden, um die Benutzer zu ermutigen, ihre Anliegen vorzubringen, und um rechtzeitige und effektive organisatorische Maßnahmen sicherzustellen.
Besondere Aufmerksamkeit sollte gerichtet werden auf Kommunikationswege, die dabei helfen, bestehende oder mögliche Quellen der Unzufriedenheit der Benutzer bildschirmgestützter Informationsverarbeitungssysteme zu erkennen und aufzugreifen.“

[11] Watzlawick, P., u. a. (1996): Menschliche Kommunikation, P. Huber Verlag

[12] Floyd, Ch., u. a. (1989): STEPS to Software Development with Users, Springer Verlag

8.6 Änderungsmanagement in Software-Entwicklungsprojekten

Der Kopf ist rund, damit das Denken die Richtung wechseln kann.

Typische Probleme, die erheblichen Einfluss auf die Nutzungsqualität haben, sind in Software-Entwicklungsprojekten die Komplexität der Anforderungen und das Managen der Änderung dieser Anforderungen während der Projektlaufzeit.

Es gibt kaum ein Projekt, in dem am Anfang dokumentierte Anforderungen unverändert bleiben. Entweder stellt die Fachseite fest, dass bestimmte Vorgaben so nicht mehr gewollt sind, neue Anforderungen hinzukommen, oder die Anforderungen von zwei Abteilungen sind nicht gemeinsam zu realisieren, oder es gibt technische Gründe, warum etwas anders realisiert werden muss als geplant, oder es stellt sich heraus, dass eine effizientere Lösung existiert usw.
In der Regel wird deshalb ein Prozess aufgesetzt, der diese Änderungen managen soll, da dieses Auswirkungen auf die finanzielle und zeitliche Planung des Projektes hat. (In Festpreisprojekten gibt es hierfür den Begriff „Change Request Verfahren". In Projekten, die nach Aufwand abgerechnet werden, ist ein solches Änderungsverfahren einfacher durchzusetzen.)

Oft gibt es bereits kurz nach Projektstart so viele Änderungsanträge, dass gerade software-ergonomische Verbesserungswünsche als der Wunsch nach so genannten „goldenen Türklinken" abgetan werden, für die man im jetzigen Stadium keine Zeit habe.
Will man software-ergonomische Aspekte wirklich in die Entwicklung integrieren, muss man einerseits am Anfang des Projektes die Anforderungen ausreichend präzise formulieren und dokumentieren, andererseits aber auch diesen Änderungsprozess mitgestalten.

Dazu gehören im wesentliche zwei Punkte:

1. **Organisation des Prozesses**
2. **Klassifizierung von Änderungsanforderungen**

Zu 1. Organisation des Prozesses:

- Beteiligte festlegen
  Es ist zu klären, wer an den Sitzungen zum Thema Änderungsmanagement teilnimmt, auch eine zeitweise Teilnahme (z. B. des Datenschutzbeauftragten) kann sinnvoll sein. Ein Benutzervertreter/Usability Engineer ist ständiges Mitglied.

- Ziele des Verfahrens
  Die Ziele sollten festgeschrieben und im Projekt kommuniziert werden. Ziele können sein:
  Koordination der zahlreichen Änderungsanforderungen, Suchen geeigneter Problemlösungen (auch kompensatorisch), Koordinieren der benötigten Ressourcen, Transparenz über Entscheidungen für alle Beteiligten.

- Aufgaben beschreiben
  Technische Kategorisierung aller eingehenden Änderungsanforderungen, z. B. Datenbankproblem, Oberflächenproblem, Netzwerkproblem, Verantwortliche für die Abarbeitung festlegen, Aufwandsschätzung, Priorisierung der Anforderungen z. B. nach sofort zu beseitigen, im nächsten Release zu beseitigen, „wenn mal Zeit ist", Umsetzungskontrolle.

- Rahmenbedingungen vereinbaren
  Regelmäßige festgelegte Termine, begrenzte zeitliche Dauer, Moderation, Dokumentation u.ä.

- Entscheidungsfindung
  Konsensorientiert und für den Fall der Nichteinigung: Eskalationsverfahren zur Projektleitung.

Obwohl es dringend zu empfehlen ist, ein Änderungsverfahren während der Entwicklung zu installieren, ist es denkbar, dass ein Änderungsverfahren, z. B. bei einem Festpreisprojekt, erst in der Nutzungsphase der Software eingesetzt wird. Denn wenn die entwickelte Software von realen Benutzern im Arbeitsalltag angewendet wird, stellen sich oft bis dahin nicht vorhersehbare Nutzungsprobleme heraus, es kommen möglicherweise „Dritte“ als Betroffene hinzu (Bürger einer öffentlichen Verwaltung) usw.

Zu 2. Klassifizierung von Änderungsanforderungen

Es gibt verschiedene Arten, Beanstandungen an einer Software zu klassifizieren. Z. B. kann nach Fehlern und Mängeln unterschieden werden.
Im deutschen Sprachgebrauch hat der Begriff „Fehler“ verschiedene Bedeutungen. Bezogen auf die Nutzungsqualität kann ein Fehler als die Nichterfüllung einer festgelegten Forderung[13] beschrieben werden. Das bedeutet, nur wenn z. B. ein ergonomisches Qualitätsmerkmal überhaupt als Anforderung dokumentiert ist, kann die Nichterfüllung als Fehler klassifiziert werden. Implizit gehören hierzu allerdings auch vorausgesetzte ergonomische Merkmale, wie Anforderungen aus Styleguides oder Normen.
Mängel hingegen werden als die „Nichterfüllung von Forderungen oder angemessenen Erwartungen bezüglich der beabsichtigten Anwendung“/des beabsichtigten Gebrauchs definiert.
So erwartet man von Software, dass vor der Ausführung von sicherheitskritischen Aktionen (z. B. Datei löschen) diese vom Benutzer noch einmal über ein Meldungsfenster bestätigt werden müssen.

Diese Begriffsdefinitionen haben allerdings mehr juristischen Charakter. Bei der Abnahme von Software durch den Auftraggeber gibt es oft eine Klassifizierung in schwere, mittlere und leichte Mängel, wobei nur schwere Mängel die Abnahme verhindern und alle software-ergonomischen Mängel als leichte oder mittlere Fehler klassifiziert werden, was bedeutet, dass die Software mit diesen Mängeln ausgeliefert wird und unter Umständen einer großen Anzahl von Nutzern jeden Tag einige Minuten Zeit zum „Drumherumarbeiten“ kostet.

Für ein konsensorientiertes Änderungsverfahren empfiehlt es sich, praxisorientiert zu klassifizieren.
Alle Änderungswünsche und Anforderungen werden in einer Tabelle aufgenommen und dann gemeinsam bewertet. Etwa in 3 Kategorien:

- Sofort zu beseitigen, da sonst kein Arbeiten möglich ist.
- In nächster Zeit zu beseitigen, d.h. innerhalb der Projektlaufzeit.
- Vorerst zu tolerieren, aber in einen Releaseplan aufzunehmen.

Für Usability-Probleme kann man folgende Kategorien definieren:

- Kosmetisches Problem (nur beseitigen, wenn genügend Zeit ist).
- Kleines Usability-Problem (geringe Priorität bei der Beseitigung).
- Usability-Katastrophe (muss unbedingt beseitigt werden).

Der Vorteil einer guten Klassifizierung ist, dass man die sortierten Probleme zu Paketen für die Planung des nächsten Release verwenden kann.

Bei der Klassifizierung von Usability-Problemen ist unbedingt zwischen Einarbeitungs- und Nutzungsproblemen zu unterscheiden.

Einarbeitungsprobleme	**typisch** • leicht voraussagbar • identifizierbar bei Erstbenutzung • vermeidbar in der (Dauer-)nutzung
(überdauernde) Nutzungsprobleme	**typisch** • schwierig voraussagbar (nur durch Aufgabenanalyse) • werden festgestellt in der (Dauer-)Nutzung • belastend in der (Dauer-) Nutzung

13 ISO 9000:2000, Beuth Verlag

Einarbeitungsprobleme sind bei der ersten Nutzung einer Software auftauchende Barrieren, die eine effiziente Erledigung der Arbeitsaufgabe erschweren, aber bei längerer Nutzung, z. B. durch gezielte Schulungs- und Unterstützungsmaßnahmen, vermeidbar sind.
Nutzungsprobleme hingegen sind Barrieren bei der Nutzung der Software, die nicht auf Einarbeitungsproble zurückzuführen sind. Diese sind zwar schwierig voraussagbar und meist erst in der (Dauer-)Nutzung festzustellen. Da solche Probleme aber objektiv belastend für die Nutzer sind und effektivitätsmindernd wirken, müssen sie eine hohe Priorität im Änderungsprozess haben.

Der nächste Schritt im Änderungsmanagement ist, zu identifizieren, wer diese Änderung vornehmen kann (in großen Projekten sind durchaus Parallelarbeiten möglich), danach muss der Aufwand für diese Änderung geschätzt werden. Hier kann es durchaus sinnvoll sein, zwei verschiedene Schätzungen einzuholen. Erfahrungsgemäß haben Entwickler manchmal kein Interesse/keine Zeit, solche Änderungen vorzunehmen und schätzen unrealistisch.

Stellt es sich heraus, dass im Änderungsgremium keine gemeinsame Einschätzung möglich ist, bietet sich das folgende Entscheidungsverfahren an: Probleme kategorisieren mit der „ABC-Analyse".

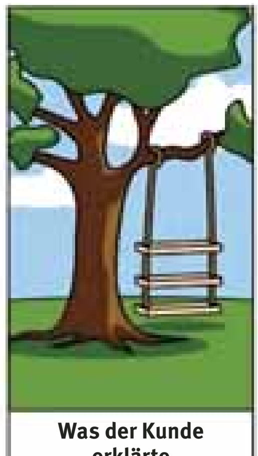
Hierzu nimmt man eine Menge ausgewählter Probleme und führt folgende Kategorisierung durch:

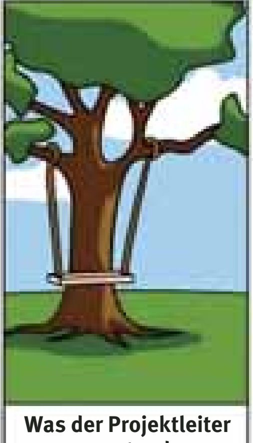
- Einige ausgewählte Fachbereichsbenutzer bewerten die Wichtigkeit der Beseitigung mit den Buchstaben (A, B, C).

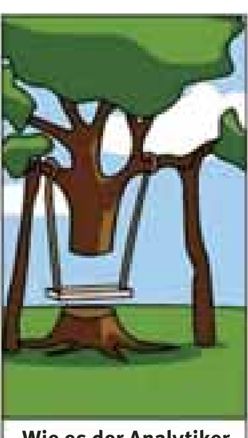
- Entwickler bewerten den Programmieraufwand mit gering, mittel und hoch (1, 2, 3).

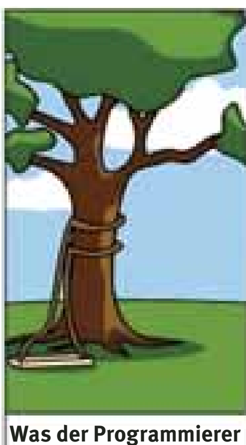
Aus der resultierenden Matrix ist eine Aufwandsschätzung und Releaseplanung ableitbar.

Ein gut organisiertes Änderungsmanagement hilft allen Beteiligten. Es verringert einerseits „unproduktive" Auseinandersetzungen andererseits sorgen konstruktive Lösungsvorschläge für eine effizientere Softwarenutzung.

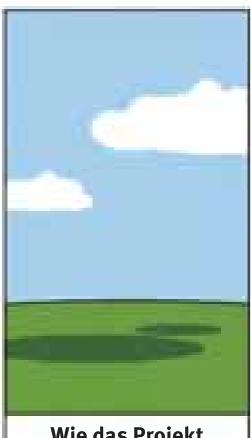

Was der Kunde erklärte

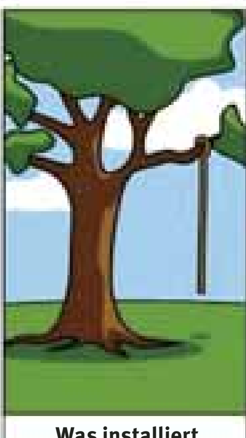

Was der Projektleiter verstand

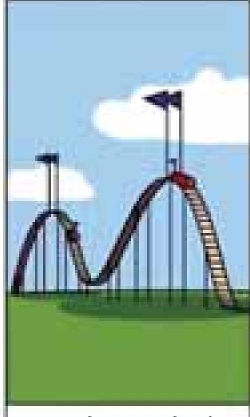

Wie es der Analytiker entwarf

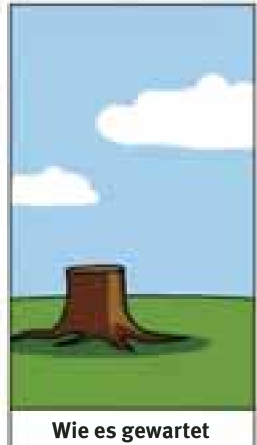

Was der Programmierer programmierte

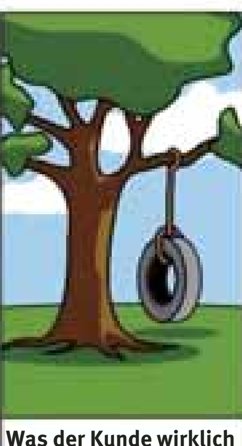

Was der Berater definierte


Wie das Projekt dokumentiert wurde


Was installiert wurde


Was dem Kunden in Rechnung gestellt wurde


Wie es gewartet wurde


Was der Kunde wirklich gebraucht hätte



9

9 Rechtliche Hintergrundinformationen

9.1 Vertragsgestaltung

9.2 Gesetzliche Grundlagen

9.3 Software-ergonomische Normen

9 Rechtliche Hintergrundinformationen

9.1 Vertragsgestaltung

Praxis ohne Theorie macht dumm.
Theorie ohne Praxis macht blind.

In optimaler Weise sollten für Software-Entwicklungen die gleichen Vertragsgrundlagen gelten wie für andere Produkte, die „entwickelt“ werden. Dies bedeutet, dass das Unternehmen/die Abteilung, die eine Software-Entwicklung in Auftrag gibt, ein Lastenheft erstellt und der Software-Hersteller ein Pflichtenheft schreibt, welches dann als Vertragsgrundlage dient. Im Lastenheft sind die Anforderungen in fachlichem Vokabular formuliert. Auf der Basis dieses Lastenheftes erstellt der Software-Hersteller ein Pflichtenheft, in dem formuliert ist, wie die fachlichen Anforderungen technisch realisiert werden können. Lastenhefte bilden deshalb die Grundlage für Ausschreibungen.
In der Praxis ist es oft der Fall, dass der Hersteller direkt das Pflichtenheft schreibt, Teile hieraus dann vom Auftraggeber nicht richtig verstanden, aber unterschrieben werden. Später kommt es dann zu Streitigkeiten. Der Auftraggeber hat in jedem Fall eine Mitwirkungspflicht, d. h. sind einzelne Aspekte nicht ausreichend detailliert beschrieben, kann er diese später nicht als Fehler bemängeln.
Gerade Nutzungsaspekte werden häufig nicht oder nur unzureichend beschrieben. Es bewirkt z. B. relativ wenig, wenn im Vertrag gefordert wird, dass die DIN EN ISO 9241-10 einzuhalten ist.
Notwendig sind möglichst spezifizierte Aussagen über die gewünschten Dialogeigenschaften. Um das Risiko einer mangelhaften Nutzungsqualität zu minimieren, sind in jedem Fall bei der Vertragsgestaltung spezifische Aussagen zu vereinbaren, dass zu einem möglichst frühen Zeitpunkt Prototypen zur Verfügung gestellt werden, und weitere Feinarbeiten erst nach deren Abnahme erfolgen. Mit Unterstützung dieser Prototypen sollten das Menü- und Navigationskonzept und das grobe Maskenlayout vereinbart werden.

Anforderungsentwicklung	Lasten-/Pflichtenheft Dienstleistungsvertrag
Programmentwicklung	Abnahmefähiges Programm Werkvertrag
Wartung und Pflege	Fehler- und Mängelbeseitigung Wartungs- und Pflegevertrag

Lastenheft
Anforderungsdefinition
Autor: Kunde/Anwender (nach Ist-Analyse)
Grundlage für eine Ausschreibung

↓

Pflichtenheft
Leistungsbeschreibung, Systemspezifikation
Autor: Software-Hersteller
Grundlage für die Abnahme

↓

Fehler- und Mängelbeseitigung sowie Wartung

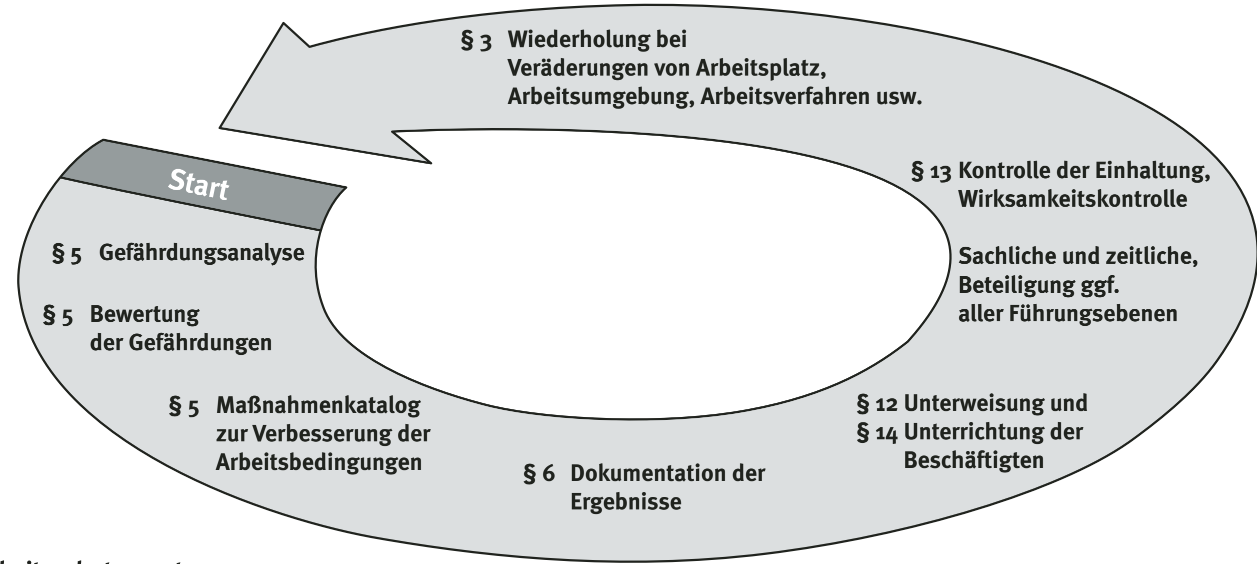
9.2 Gesetzliche Grundlagen

9.2.1 Das Arbeitsschutzgesetz (ArbSchG)

Bedeutende Bestandteile des Arbeitsschutzgesetzes sind

- Gesundheitsbegriff:
  Zustand psychischen, geistigen und sozialen Wohlbefindens
- Ganzheitlichkeit
- Mensch steht im Mittelpunkt
- Zusammenwirken verschiedener Faktoren berücksichtigen
- Ziel ist nicht allein die Vermeidung von Arbeitsunfällen, sondern die Gesunderhaltung der Arbeitnehmer.

Die im Arbeitsschutzgesetz vorgeschriebenen Gefährdungsbeurteilungen müssen je nach Art des Arbeitsplatzes durchgeführt, dokumentiert und bei Änderungen wiederholt werden.



***Arbeitsschutzgesetz
Erforderliche Maßnahmen mit dem Ziel der
Verbesserung von Gesundheit und Sicherheit***

9.2.2 Die Bildschirmarbeitsverordnung (BildscharbV)

Durch die 1996 im Zusammenhang mit dem Arbeitsschutzgesetz in Kraft getretene Bildschirmarbeitsverordnung hat die Software-Ergonomie eine besondere Bedeutung bekommen.

Aus der Bildschirmarbeitsverordnung (Anhang Punkt 21) geht hervor, dass zukünftig eine prozessartige Verbesserung der software-ergonomischen Qualität von Anwendungssoftware anzustreben ist.
Die Handlungsmöglichkeiten während der einzelnen Phasen des Software-Lebenszyklus sind der Abbildung zu entnehmen.

Die im Arbeitsschutzgesetz vorgeschriebenen Analysen müssen sich bei Bildschirmarbeitsplätzen insbesondere auf folgende Faktoren konzentrieren:

- Sehvermögen,
- körperliche Probleme und
- psychische Belastungen.

Im Anhang der Verordnung finden sich dann so genannte Mindestvorschriften:

- Die Grundsätze der Ergonomie sind insbesondere auf die Verarbeitung von Informationen durch den Menschen anzuwenden.

Anmerkung: Die „Grundsätze der Ergonomie“ sind in den Normen (insbesondere DIN EN ISO 9241-12) beschrieben.

- Bei Entwicklung, Auswahl, Erwerb und Änderung von Software sowie bei der Gestaltung der Tätigkeit an Bildschirmgeräten hat der Arbeitgeber den folgenden Grundsätzen insbesondere im Hinblick auf die Benutzerfreundlichkeit Rechnung zu tragen:

Anmerkung: Dies bedeutet, dass die Grundsätze eingehalten werden müssen, unabhängig davon, ob es sich um die Auswahl einer Standardsoftware oder Eigenentwicklungen handelt. Darüber hinaus ist die ergonomische (DIN EN ISO 9241-2) Aufgabengestaltung (im Zusammenhang mit der Software-Nutzung) zu berücksichtigen.

Zur Software ist detaillierter beschrieben

- Die Software muss an die auszuführende Aufgabe angepasst sein.

- Die Systeme müssen den Benutzern Angaben über die jeweiligen Dialogabläufe unmittelbar oder auf Verlangen machen.

- Die Systeme müssen den Benutzern die Beeinflussung der jeweiligen Dialogabläufe ermöglichen sowie eventuelle Fehler bei der Handhabung beschreiben und deren Beseitigung mit begrenztem Arbeitsaufwand erlauben.

- Die Software muss entsprechend den Kenntnissen und Erfahrungen der Benutzer im Hinblick auf die auszuführende Aufgabe angepasst werden können.

- Ohne Wissen der Benutzer darf keine Vorrichtung zur qualitativen oder quantitativen Kontrolle verwendet werden.

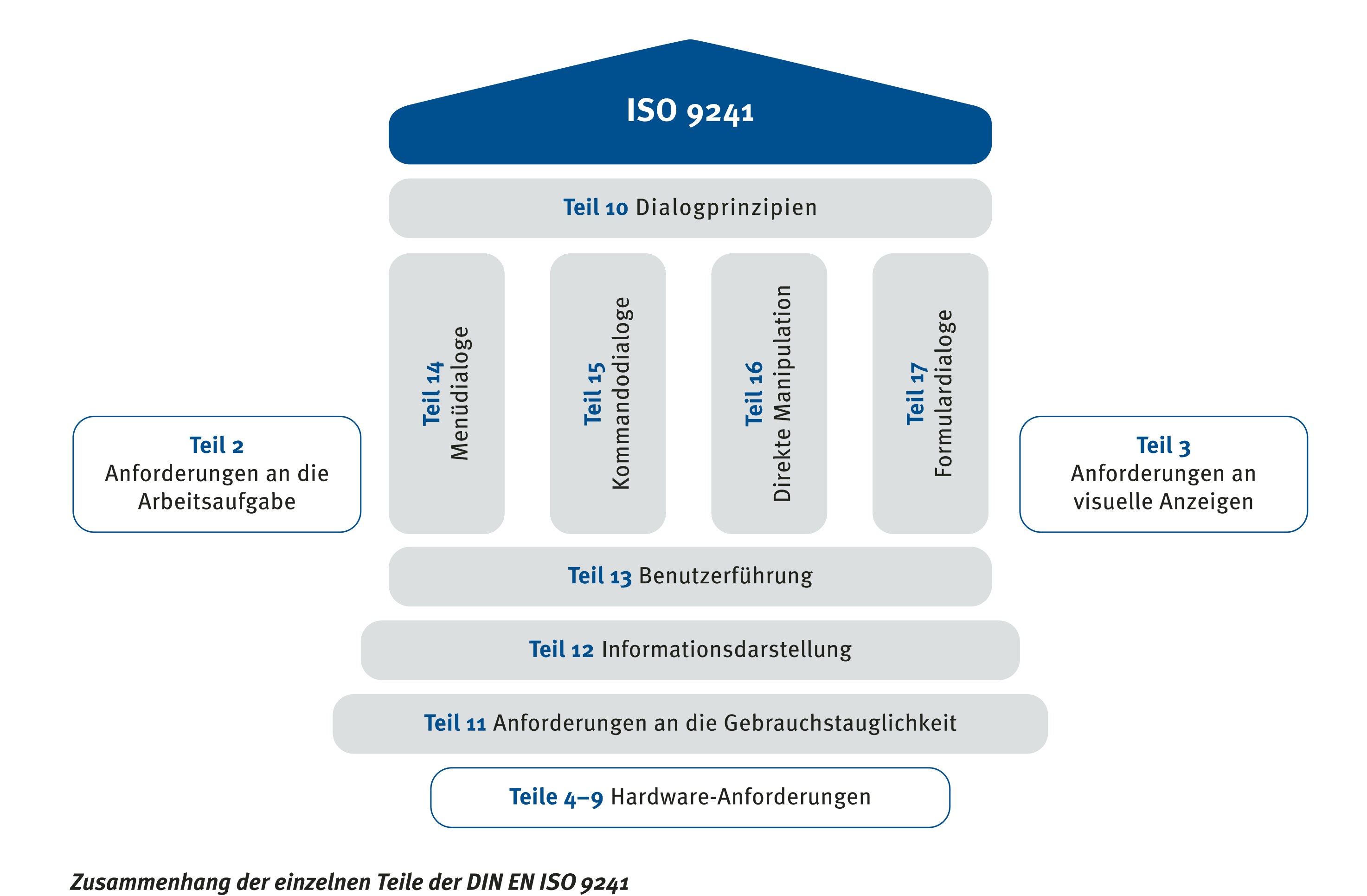
9.3 Software-ergonomische Normen

Es gibt zahlreiche internationale Normen zur Bildschirmarbeit. Diese werden ins europäische und deutsche Normenwerk übernommen, was zu der Bezeichnung DIN EN ISO führt (DIN = Deutsches Institut für Normung, EN = europäische Normung, ISO = International Organization for Standardization).

Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten

Basierend auf den Arbeiten von Dzida, Herda und Itzfeld (1978) entstand vor fast 30 Jahren die erste deutsche Norm für benutzungsfreundliche Software, die DIN 66234-8, die 1988 veröffentlicht wurde.
Diese deutsche Norm wurde 1996 dann die Grundlage für den Teil 10 des europäischen und internationalen Standards DIN EN ISO 9241 (Ergonomic requirements for office work with display terminals VDTs).
Die DIN EN ISO 9241 besteht z. Z. aus insgesamt 17 Teilen.

Die DIN EN ISO 9241 dient der Konkretisierung ergonomischer Anforderungen. Dies bedeutet, dass sie zur Auslegung der Bildschirmarbeitsverordnung herangezogen werden muss. Dies bedeutet darüber hinaus, dass das, was in den Normen steht, Mindestanforderungen sind, die erfüllt sein müssen. Software-ergonomische Normen unterscheiden sich jedoch von technischen Normen. In technischen Normen werden Produkteigenschaften genormt. Software-ergonomische Normen haben Richtliniencharakter. Dieser Richtliniencharakter ist sinnvoll, weil Gestaltungsspielraum bleibt, da es keine Richtlinien für ein bestimmtes Produkt sind und es nicht den einen Benutzer-Typ gibt. Darüber hinaus ist es so möglich, neue Erkenntnisse anzuwenden. Denn über die Jahre, die eine Norm gilt, bevor sie überarbeitet wird, ändern sich die technisch-inhaltlichen Anforderungen, deshalb finden sich in den Normen abstrakte Gestaltungsprinzipien (statt Gestaltungsregeln), die für jeden Anwendungsfall konkretisiert werden müssen. Demzufolge können software-ergonomische Normen nur am Arbeitsplatz geprüft werden (im Gegensatz zu technischen Normen: ein Bildschirm kann in einem Labor getestet werden).



Zusammenhang der einzelnen Teile der DIN EN ISO 9241

Die DIN EN ISO 9241 beinhaltet weit über 400 Gestaltungsregeln. Dies bedeutet jedoch nicht, dass diese bei einem Softwareprodukt alle erfüllt sein müssen. Software-Entwicklung ist ein kreativer Prozess. Auch wenn Entwickler von den Normtexten enttäuscht sind, so sind sie doch nicht anders zu realisieren. Entwickler erwarten oft exakte Entscheidungsregeln oder Checklisten. Dies würde aber nicht funktionieren. Immer ist Rückkopplung mit den Benutzern und deren Arbeitsaufgaben notwendig.
Diese Ableitung der Anforderungen führt dazu, dass entschieden werden kann, ob eine Gestaltungsregel anzuwenden ist oder nicht.
Zur Erläuterung ein Beispiel aus der DIN EN ISO 9241 Teil 10. Dort wird zum Kriterium Aufgabenangemessenheit als Beispiel beschrieben:

- „Gibt es für eine Arbeitsaufgabe Standardwerte, sollten diese dem Benutzer als Vorgabe angeboten werden. Die Werte sollen durch andere Werte oder andere Vorgaben ersetzt werden können.“

Es wäre also zu prüfen, ob der Arbeitsablauf die Eingabe von Standardwerten erfordert. Ist dies der Fall, muss diese Vorschrift angewendet werden. Es handelt sich jedoch nur um ein Beispiel, was bedeutet, dass die Kriterien, die für die „eigene“ Software gelten sollen, konkretisiert, das heißt ausformuliert werden müssen. Dies erfolgt vorzugsweise zu Beginn des Projektes, damit dies am Ende als Qualitätsmerkmal geprüft werden kann. Denn ist nichts festgelegt, ist die Gefahr von Auslegungsstreitigkeiten gegeben.

Unkomplizierter ist es bei Vorschriften aus Teil 12 der Norm:

Bei der Informationsdarstellung ist als Beispiel für die Darstellung numerischer Information in Listen beschrieben:

- Numerische Informationen ohne Dezimalzeichen (Punkt, Komma) sollten rechtsbündig angeordnet werden.

- Numerische Informationen mit Dezimalzeichen sollten unter Berücksichtigung des Zeichens ausgerichtet werden.

Diese Regel ist ebenfalls nur anzuwenden, wenn Auflistungen vorhanden sind, und stellt einen sinnvollen praktischen Gestaltungsvorschlag dar.

Aus rechtlichen Gründen dürfen die Normen hier nicht abgedruckt werden. Man kann diese nur kostenpflichtig beim Beuth Verlag bestellen. Zitate aus den Normen findet man in dieser Broschüre und zahlreichen anderen Veröffentlichungen oder im Internet.
Nicht eingegangen ist diese Broschüre auf die Teile 15 und 16. Es handelt sich im Teil 15 um Festlegungen zur Verwendung von Kommandosprachen. Das heißt, statt einer Aus-wahl gibt der Benutzer einen Befehl oder eine Folge von Befehlen ein. Diese Methode der Dialogführung wird oft im sicherheitskritischen Bereich verwendet, z. B. auch bei zeichenorientierten Anwendungen. Hier finden sich in der Norm Hinweise zu Struktur und Syntax, Darstellung der Befehle, Ein- und Ausgabebetrachtungen, Rückmeldungen und Hilfe. Im Teil 16 sind Regeln für die Dialogführung mittels direkter Manipulation beschrieben, die Interaktionsform, die man von grafischen Oberflächen kennt, z. B. wenn ein Dokument mit der Maus in den Papierkorb geschoben wird. In diesem Normteil sind Richtlinien zur grundsätzlichen Verwendung von Metaphern (Desktop-Oberfläche, z. B. Nutzung des Papierkorbs ist kein grundsätzlicher Datenverlust) beschrieben, wie die Manipulation grafischer Objekte dem Benutzer rückgemeldet wird (Änderung der Cursor-Form zeigt dem Benutzer an, was er machen kann), zur Art der Eingabe und zur Manipulation von Objekten.

Derzeit wird ein neuer Usability-Standard erarbeitet, dessen Arbeitstitel „ISO 23973 Software Ergonomics for World Wide Web User Interfaces“ lautet. Die Norm ISO 23973 greift die Grundgedanken der ISO 13407 auf (nutzerorientierte, iterative Gestaltung interaktiver Systeme: human-centered Designprozesse) und richtet sich vor allem an Web-Entwicklerteams (und damit gleichermaßen an Konzepter, Designer, Programmierer, Projektmanager und Usability-Spezialisten).

ISO 23973 stützt sich auf ein Referenzmodell, welches den Webdesign-Prozess in drei so genannte Domänen aufgliedert, die sich gegenseitig bedingen: Design-, Evaluations- und Prozess-Domäne. Letztere bezieht sich auf die nutzerorientierten Design-Prozesse wie in ISO 13407 beschrieben. Die Evaluations-Domäne beinhaltet den kompletten Werkzeugkasten rund um Verfahren, Methoden und Instrumente wie Qualitätssicherung und Usability-Testing. ISO 23973 selbst konzentriert sich in erster Linie auf die Aspekte der Design-Domäne bzw. die unterschiedlichen Levels im Designprozess: Ziel, Zweck und Strategie/Content und Funktionalität/Navigation und Interaktion/Präsentation und Mediendesign.

1.1 Gesetze und Normen

Arbeitsschutzgesetz – ArbSchG (7.8.1998): Bundesgesetzblatt I, S. 1246.
Zuletzt geändert am 19.12.1998, Bundesgesetzblatt I, S. 3843
http://www.gesetze-im-internet.de/index.html

Bildschirmarbeitsverordnung – BildscharbV (20.12.1998)
http://www.gesetze-im-internet.de/index.html

Normen :
DIN. Deutsches Institut für Normung. Alle deutschsprachigen Normen beziehbar über den Beuth Verlag, Berlin. http://www.beuth.de/.

DIN EN ISO 9241-110 Ergonomie der Mensch-System-Interaktion, 2006
Teile 2, 3, 10–17

ISO 14915 2003-04 Software-Ergonomie für Multimedia-Benutzungsschnittstellen

DIN EN ISO 13407 2000-11 Benutzer-orientierte Gestaltung interaktiver Systeme.

ISO 9000:2000, Beuth Verlag. http://www.beuth.de

ISO/IEC 12119:1995-03 Informationstechnik. Software-Pakete. Qualitätsanforderungen und Prüfung

DATech (2006): Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die softwareergonomische Evaluierung von Software auf der Grundlage von DIN EN ISO 9241, Version 3.4.
Teile 10 und 11. http://www.datech.de (Verfahren & Unterlagen › Prüflaboratorien)

DATech (2006) Prüfbaustein Usability Engineering-Prozess. Leitfaden für die Evaluierung des Usability Engineering-Prozesses bei der Herstellung und Pflege von Produkten auf der Grundlage von DIN ISO 13407 Version 1.4 http://www.datech.de (Verfahren & Unterlagen › Prüflaboratorien)

1.2 Tagungen, Institutionen

http://www.datech.de/

http://www.mensch-computer-interaktion.de/ (Fachbereich der Gesellschaft für Informatik)

http://www.mensch-und-computer.de (Jahrestagungs-Homepage)

http://www.sozialnetz-hessen.de/ergo-online (Online-Angebot aus Sicht von Gewerkschaften)

1.3 Usability-Seiten von Software-Herstellern

Microsoft: http://www.microsoft.com/Usability/

IBM: http://www-5.ibm.com/de/kn/tu/tu_info9_usability.html

SAP: http://www.sapdesignguild.org/index.asp

1.4 Styleguides (Beispiele)

http://www.sapdesignguild.org./resources/MiniSG/index.htm

http://java.sun.com/products/jlf/ed1/dg/index.htm

http://www.webstyleguide.com/

1.5 Software-Produktgestaltung

Aicher, Krampen: Zeichensysteme der visuellen Kommunikation, 1996 Ernst & Sohn Verlag

Balzert, H.: Webdesign & Ergonomie, 2004 W3L Verlag, Herdecke

Görner, C., Beu, A.,Koller, F.: Der Bildschirmarbeitsplatz, Softwareentwicklung nach der DIN EN ISO 9241, 1999 Beuth Verlag

Heinecke, A.: Mensch-Computer-Interaktion, 2004 Fachbuchverlag Leipzig, http://www.drheinecke.de

Herczeg, M.: Software-Ergonomie, 2005 Oldenburg

Nielsen, J.: Designing Web Usability, dtsch. Ausg. 2004 Markt und Technik

Norman, D.: Dinge des Alltags, 1988 Campus

Puscher, F.: Das Usability Prinzip-Wege zur benutzungsfreundlichen Website, 2001 dpunkt.verlag

Richenhagen, Prümper, Wagner: Handbuch der Bildschirmarbeit, 1997 Luchterhand

Staufer: Piktogramme für Computer, 1987 Gruyter

Thissen, F.: Screen Design Handbuch, 2000 Springer

Ulich, E.: Arbeitspsychologie, 2001 Schäffer Poeschel

Wirth, T.: Missing Links, 2004 Hauser Fachbuchverlag

1.6 Software-Entwicklungsprozess

Beyer, H., Holtzblatt, K.: Contextual Design Defining Customer-Centred Systems, 1998 Morgan Kaufmann Publishers, Inc.

Bias, R. G. and Mayhew, D. J.: Cost-Justifying Usability, 1994 Academic Press, Boston

Heinsen, S., Vogt, P.: Usability praktisch umsetzen, 2003 Hanser Fachbuchverlag

Landauer, T. K.: The Trouble with Computers, 1995 The MIT Press, Cambridge MA, London

Mayhew, D. J.: The Usability Engineering Lifecycle. A Practitioner's Handbook for User Interface Design, 1999 Morgan Kaufmann, San Francisco

Nielsen, J.: Usability Engineering, 1993 Academic Press

Richenhagen, G., Prümper, J. und Wagner, J.: Handbuch der Bildschirmarbeit, 1997 Luchterhand

Shneidermann, B.: Workshop on Usability Testing, 1998 NIST, 3.5.1998 SIGCHI Bulletin

Unfallkasse Post und Telekom
Europaplatz 2, 72072 Tübingen

Unser **Service-Center** hilft Ihnen
schnell und kompetent weiter
Telefon: 0180 5001632
oder 07071 933-0
Montag bis Donnerstag
von 8:00-16:00 Uhr
Freitag von 8:00-14:30 Uhr
Fax: 07071 933-4399
E-Mail: info@ukpt.de
Internet: www.ukpt.de

2. überarbeitete Auflage

2006 / MatNr 670-095-361

Software-Ergonomie



